# TIFF-Korpus-Analyse

**Martin Kaiser, Claire Röthlisberger-Jourdan, Georg Büchler**

## Ausgangslage

TIFF ist gegenwärtig und seit langem das meistgebrauchte Format zur Archivierung von unkomprimierten Bilddaten.[1] Es handelt sich um ein flexibles, anpassungsfähiges Dateiformat, das über die Jahre eine Vielzahl von Erweiterungen und Ergänzungen erfahren hat. Daneben bietet es die Möglichkeit, Metadaten in anderen Standards (wie IPTC, EXIF oder ICC) einzubetten. Diese Flexibilität und Ausprägungen machen TIFF jedoch zu einem komplexen Dateiformat und bergen Risiken für die digitale Archivierung. Die Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen KOST hat deshalb bereits 2014 eine Empfehlung zum Preservation Planning für TIFF-Dateien publiziert, welche basierend auf der Baseline-TIFF-Spezifikation ein archivtaugliches TIFF zu definieren versucht.[2] 2015 bis 2017 strebte ein gemeinsames Projekt[3] des *Digital Humanities Lab* an der Universität Basel (DHLab), der *Universität Girona* und der Firma *Easy Innova* an, eine erweiterte Baseline-Spezifikation in eine *ISO Recommendation* zu überführen, um so dem Umstand abzuhelfen, dass TIFF eine offene Spezifikation der Firma Adobe, jedoch kein ISO-Standard ist.[4]

Damit eine solche Empfehlung nicht nur auf theoretischen Überlegungen beruht, sondern sich auf eine fundierte Analyse echter archivischer Daten stützen kann, haben es die KOST und das DHLab im Rahmen dieses Projekts unternommen, mehrere Millionen Dateien aus drei Archiven systematisch zu untersuchen. Parallel dazu wurden an diesem Korpus auch etliche bekannte und in der Archivwelt verbreitete Analysetools getestet.

---

1 Dieser Beitrag ist eine überarbeitete Version des Artikels «TIFF-Korpus-Analyse» auf der KOST-Website, https://kost-ceco.ch/cms/index.php?tiff-data-analysis_de. Wir danken den Kollegen in den beteiligten KOST-Trägerarchiven für ihre Mitarbeit an der Korpus-Analyse: Marcel Büchler (Schweizerisches Bundesarchiv); Martin Lüthi und Vedat Akgül (Staatsarchiv St. Gallen); Markus Loch und Lambert Kansy (Staatsarchiv Basel-Stadt). Unser Dank geht ebenfalls an das Projektteam des TI-A-Projekts, im Besonderen an Peter Fornaro und Erwin Zbinden vom Digital Humanities Lab der Universität Basel. (Sämtliche Weblinks wurden am 19.02.2018 zuletzt aufgerufen.)

2 https://kost-ceco.ch/cms/index.php?preservation_tiff_de.

3 http://ti-a.org/.

4 Ein entscheidender Nachteil dieser Konstellation zeigt sich seit Ende 2016: Die aktuelle TIFF-Spezifikation (Version 6.0), früher publiziert unter http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf, ist kommentarlos von der Adobe-Website verschwunden. Sie bleibt an anderen Quellen greifbar, zum Beispiel beim Internet Archive unter https://web.archive.org/web/20091223030231/http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf.

Informationswissenschaft: Theorie, Methode und Praxis, Bd. 5 (2018), Nr. 1 — http://dx.doi.org/10.18755/iw.2018.8

## Korpus und Fragestellungen

### TIFF-Korpus

Die Staatsarchive Basel-Stadt und St. Gallen und das Schweizerische Bundesarchiv stellten für die Untersuchung ihre TIFF-Sammlungen von je etwa 12 TB zur Verfügung. Die Bestände sind, was Alter, Grösse und Ursprung betrifft, in allen Archiven sehr heterogen. Eine summarische Aufstellung zeigt Tabelle 1.

| **Staatsarchiv Basel-Stadt** | | |
|---|---|---|
| Typ | Anzahl | Grösse |
| 2000 | 1‘950 | |
| 2001 | 2‘300 | |
| 2002 | 200 | |
| 2003 | 100 | |
| 2004 | 10‘000 | |
| 2005-2015 | 750‘000 | |
| **Total** | **764‘550** | **12.4 TB** |
| **Staatsarchiv St. Gallen** | | |
| | Anzahl | Grösse |
| **Total** | **870‘000** | **12.83 TB** |
| **Schweizerisches Bundesarchiv** | | |
| Typ | Anzahl | Grösse |
| Archiv-TIFFs | 1‘700‘000 | 5-6 TB |
| Digitalisate | 7300 | 0.46 TB |
| Digitalisate von Dritten | 14‘000‘000 | 6 TB |
| **Total** | **15‘707‘300** | **11-12 TB** |

**Tabelle 1: Anzahl und Grösse der analysierten TIFF-Dateien**

### Fragestellungen und Analyseprogramme

Für die Analyse der TIFF-Dateien wurden die folgenden Programme ausgewählt:

| | |
|---|---|
| MD5-Berechnung | MD5<br>Das Berechnen des MD5-Schlüssels gehört nicht zu den Analysemodulen, garantiert aber die Lesbarkeit der Datei. |
| Formaterkennung | file<br>http://gnuwin32.sourceforge.net/packages/file.htm<br>Mit der Formaterkennung durch file werden falsch gekennzeichnete Dateien erkannt. |
| Formatvalidierung | JHOVE<br>http://jhove.openpreservation.org/ |

| | |
|---|---|
| | Die JHOVE-Validierung ermittelt die grundlegende Struktur der TIFF-Datei. Wichtig sind hier Status und InfoMessage. |
| Formatvalidierung | DPF-Manager<br>http://www.preforma-project.eu/dpf-manager.html<br>Der DPF-Manager ist eine Alternative zu JHOVE aus dem PREFORMA-Projekt. |
| Validierung und TIFF-Tag-Extraktion | checkit_tiff: a conformance checker for baseline TIFFs<br>https://github.com/SLUB-digitalpreservation/checkit_tiff<br>checkit_tiff wurde von der Sächsischen Landesbibliothek – Staats- und Universitätsbibliothek Dresden entwickelt. |
| TIFF-Tag-Extraktion | tiffhist<br>http://dhlab.unibas.ch/<br>Das vom DHLab entwickelte C++-Programm extrahiert alle TIFF-Tags in eine CSV-Tabelle (TIFF-Tag, Datentyp und Wert) |
| EXIF-Extraktion | ExifTool (EXIF-Extraktion)<br>http://owl.phy.queensu.ca/~phil/exiftool/<br>Eingebettete EXIF- und XMP-Metadaten werden extrahiert. |
| Thumbnail-Generierung | ImageMagick<br>http://www.imagemagick.org/<br>Für jede Datei wird mit ImageMagick ein sehr kleines Thumbnail generiert. In diesem Schritt wird somit die Payload oder Bitmap der TIFF-Datei untersucht. Eine erfolgreiche Konvertierung belegt die korrekte Implementierung von Komprimierung und Farbraum. |

**Tabelle 2: Verwendete Analyseprogramme**

## Vorgehen im Detail

Weil davon auszugehen war, dass viele Dateien noch einer Schutzfrist unterstehen oder urheberrechtlich geschützt sind, sollten die zu untersuchenden Korpora die beteiligten Archive nicht verlassen. Auch sollten über Dateinamen oder Pfadnamen keine Rückschlüsse auf die Archivbestände gezogen werden können. Deswegen wurden in einem ersten Schritt die TIFF-Dateien aus dem jeweiligen Archivsystem auf USB- oder NAS-Platten kopiert und diese anschliessend für die weitere Untersuchung vom Netzwerk des jeweiligen Archivs abgehängt. Das Kopieren nahm wegen organisatorischer Herausforderungen und wegen der Datenmenge etwa 3 Monate in Anspruch.

Um die Anforderungen an den Umgang mit grossen Datenmengen, langen Programmlaufzeiten und sicherer Anonymisierung erfüllen zu können, wurde beschlossen, Programmausführung und Logverwaltung mit einer Datenbank und einem speziellen Analyse-Loop-Programm zu realisieren. Dieses musste sowohl im Linux- als auch im Windows-Umfeld eingesetzt werden können. Ein Abbruch in einem Analyseschritt durfte die nächsten Tools nicht beeinflussen. Als Datenbank wurde SQlite und als Programmiersprache für das Überwachungsprogramm Golang

gewählt.[5] SQlite hat den Vorteil, dass weder Server noch Administration notwendig sind. Golang ist eine kompilierte Sprache und auf allen Plattformen verfügbar, hat ein API zu SQLite und ist einfacher als C/C++. Alle Programme, Datenbankmodelle, Scripts und SQL-Abfragen sind auf GitHub verfügbar; eine detaillierte Installationsanleitung findet sich auf der KOST-Website.[6]

Diese Konstellation erlaubte es, die Ausführung der Analysemodule von der Auswertung der Log- oder Systemausgabe vollständig zu trennen. Die Log- oder Systemausgabe zu jedem Analyseschritt wurde für die spätere Auswertung festgehalten. Um der Anforderung der vollständigen Anonymisierung gerecht zu werden, wurden die Logdateien beim Schreiben gefiltert und Pfad und Dateinamen entfernt. Die eigentliche Auswertung erfolgt anschliessend vollständig offline, entweder im Archiv oder ausgelagert. Die Vorteile dieses Vorgehens sind, dass verschiedene Auswertungen auch zeitlich versetzt möglich sind und dass die Fragestellungen und Auswertungsmethoden während der Arbeit noch verändert werden können. Für die Auswertung stehen nach der Analyse insgesamt etwa 35 GB Log-Informationen zur Verfügung.

### Analyse-Loop-Programm

Das Analyse-Loop-Programm liest alle TIFF-Dateien des Korpus vom NAS und führt mit der jeweils gelesenen Datei durch Aufrufen von externen Programmen mehrere Analyseschritte aus. Der Loop-Prozess ist zweiteilig und besteht aus der Initialisierung der Prozessdatenbank und der eigentlichen Analyse.

Der Initialisierungsschritt (Abbildung 1) erstellt die Datenbank und schreibt für jede TIFF-Datei einen Eintrag mit dem Pfad und Dateinamen als Schlüssel. Die Initialisierung kann mehrfach aufgerufen werden und fügt so neue Verzeichnispfade zur Datenbank hinzu. Um eine spätere anonymisierte Auswertung ausserhalb der Archive zu ermöglichen, werden Dateinamen und Dateipfad, welche allenfalls Rückschlüsse auf den Inhalt der Dateien erlauben würden, in einer separaten Tabelle (*namefile*) gehalten.

5 Siehe https://www.sqlite.org/ und https://golang.org/.

6 Siehe https://github.com/KOST-CECO/TiffAnalyseProject beziehungsweise https://kost-ceco.ch/cms/index.php?tiff-data-analysis_de.

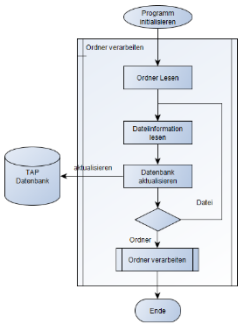


**Abbildung 1: Initial Loop liest sämtliche TIFF-Dateien**

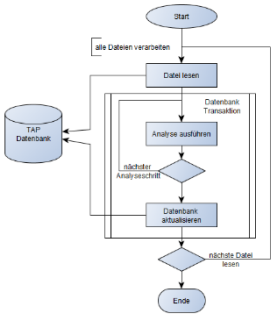


**Abbildung 2: Process Loop führt Analyseschritte aus**

Im Analysefall (Abbildung 2) werden die Dateieinträge in der Datenbank abgearbeitet und die Analysetools im Kommandozeilenmodus aufgerufen. Durch die Verwendung einer Datenbank ist jederzeit ein Abbrechen und Neustarten der Analyse möglich. Die Analyse umfasst folgende Schritte:

- Der Dateipfad und der Pfad zur Logdatei werden dem Analyse-Loop-Programm im Kommandozeilenmodus übergeben.
- Falls das Analyse-Loop-Programm kein Logfile im Append-Modus öffnen kann, hängt es den Log-Output entweder an die Logdatei an oder schreibt ihn in die Datenbank.
- Der aktuelle Offset der Logdatei wird in der Datenbank gespeichert.
- Der Exitstatus des Analysetools wird in der Datenbank festgehalten.
- Es kann festgelegt werden, ob der Systemoutput des Analysetools in eine spezielle Ausgabedatei geschrieben oder in der Datenbank gespeichert werden soll.
- Eine Logrotation verhindert allzu grosse Logdateien.

## Datenmodell der Analysedaten

Die Tabellen *keyfile* und *namefile* enthalten den primären Verzeichnisscan, also die Namen aller Dateien mit Dateigrösse und Erstellungszeit, soweit diese aus dem Lesen der Verzeichnisstrukturen erstellt werden können. Zum Ausführen der Analysemodule werden die notwendigen Informationen aus der Tabelle *analysetool* ausgelesen: Programmname und Pfad, Logdatei, Datei bzw. BLOB für den Systemoutput. Die Tabelle *status* hält den Exitstatus des Analyseprogramms fest. Die Tabellen *logindex* und *sysindex* speichern entweder den Dateinamen und den Offset in die jeweilige Logdatei oder den gesamten Output der eben analysierten Datei in ein entsprechendes LOB.

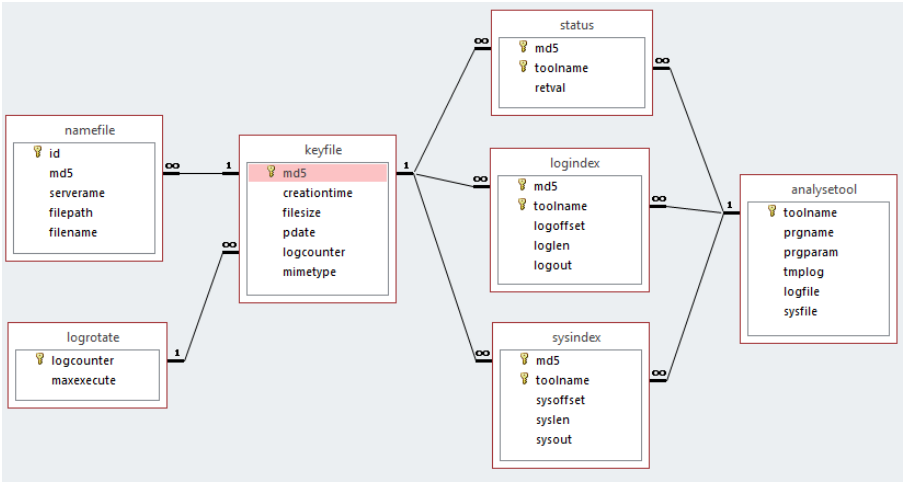


**Abbildung 3: Das Datenmodell in grafischer Darstellung**

| tablename | name | description |
|---|---|---|
| analysetool | toolname | Name des registrierten Analyseprograms in Kurzform |
| | prgname | Pfad und Dateiname zum Analyseprograms |
| | prgparam | Parameter des Analyseprogramms mit Wildcards %file% und %log% |
| | tmplog | Temporäre Logdatei: ersetzt Wildcards %log% beim Ausführen des Analyseprogramms, Fehlen meint keine Logdatei schreiben |
| | logfile | Pfad und Dateiname der mit diesem Analyseprogramms verbunden Logdatei: Ist kein Logfile definiert, wird in LOB «logout» gespeichert |
| | sysfile | Pfad und Dateiname der mit diesem Analyseprogramms verbunden Ausgabedatei: Ist kein Sysfile definiert, wird in LOB «sysout» gespeichert |
| | | |
| keyfile | md5 | MD5-Hashwert und Referenz zum namefile |
| | creationtime | Entstehungszeitpunkt der Datei laut Dateisystem |
| | filesize | Dateigrösse in Byte |

| | | |
|---|---|---|
| | pdate | Zeitpunkt und Flag für den Abschluss der gesamten Analyse |
| | logcounter | Zähler für «logfile» bzw. «sysfile» beginnend mit Eins |
| | mimetype | MIME Type (Internet Media Type) der Datei gemäss der Magic Number |
| | | |
| logindex | md5 | MD5-Schlüssel der TIFF-Datei |
| | toolname | Kurzname des Tools |
| | logoffset | Offset in die Ausgabedatei analysetool.logfile |
| | loglen | Länge des Logausgabe |
| | logout | Vollständige Logausgabe des Analysetools oder Name der Logdatei |
| | | |
| logrotate | logcounter | Zähler für «logfile» bzw. «sysfile» beginnend mit eins |
| | maxexecute | Maximale Verarbeitungsschritte pro «logfile» bzw. «sysfile» |
| | | |
| namefile | id | Referenz zu «keyfile» |
| | md5 | MD5-Hashwert |
| | servername | Name des NAS-Servers oder des zugeordneten Laufwerkbuchstabens |
| | filepath | Dateipfad |
| | filename | Dateiname mit Dateiextension |
| | | |
| status | md5 | MD5-Schlüssel der TIFF-Datei |
| | toolname | Kurzname des Tools |
| | retval | Rückgabewert des Tools[7] |
| | | |
| sysindex | md5 | MD5-Schlüssel der TIFF-Datei |
| | toolname | Kurzname des Tools |
| | sysoffset | Offset in die Ausgabedatei analysetool.sysfile |
| | sylen | Länge der Konsolenausgabe |
| | sysout | Vollständige SystemOut-Ausgabe des Analysetools (stderr und stdout) oder Name der Logdatei |

**Tabelle 3: Das Datenmodell mit Tabellen**

## Einschränkungen bei der Analyse

Nach den ersten Tests hat sich schnell gezeigt, dass die verwendeten Tools sehr unterschiedliche Rechenzeiten pro TIFF-Datei erfordern. Tabelle 4 dokumentiert die Rechenzeiten pro Tool über 1000 Dateien unterschiedlicher Grösse (mittlere Grösse ~5.5 MB), indexiert gegenüber *tiffhist*:

7 Siehe dazu http://www.hiteksoftware.com/knowledge/articles/049.htm.

| Tool | Zeit (s) | Faktor |
|---|---|---|
| tiffhist | 257 | 1.0 |
| dpf-manager | 257 | 1.0 |
| file | 266 | 1.0 |
| exiv2 | 267 | 1.0 |
| exif | 335 | 1.3 |
| checkit_tiff | 503 | 2.0 |
| jhove | 697 | 2.7 |
| ImageMagick | 3424 | 13.3 |
| Total | 6006 | 23.4 |

**Tabelle 4: Rechenzeiten pro Tool über 1000 Dateien unterschiedlicher Grösse**

Um die Analysezeit nicht ausufern zu lassen, wurde beschlossen, ImageMagick nur über einem sehr kleinen Teilbestand und JHOVE nur etwa über der Hälfte der Dateien auszuführen. Das aus unserer Sicht wichtigste Tool zur Extraktion von TIFF-Tags, das vom DHLab entwickelte *tiffhist*, wurde hingegen auf allen Dateien ausgeführt.

## Auswertung

Die Analyseresultate stehen der gesamten Fachgemeinschaft zur Verfügung. Sie sind zu diesem Zweck auf der Website der KOST publiziert und erläutert[8]. Alle Interessierten sind eingeladen, diese Daten für ihre eigenen Forschungen zu benützen; besonders angesprochen sind dabei die Hochschulen und Fachhochschulen.

Abschliessend folgen hier zwei ausgewählte Beispiele für eine mögliche Auswertung. Weitere Beispiele sind auf der KOST-Website dokumentiert.

8 Siehe https://kost-ceco.ch/ftp_space/TIFF-Analyse/. Der Downloadspace enthält die Analysedatenbank als SQL Loader Script tap.sql.gz (md5: 33b406a083472fb3853c1d07169bd640) und die Logdateien in einem TAR-File log.tgz (d44b169f1d8048637c5502be646f8a85). In separaten Dateien ist die später fertiggestellte Analyse der 10 Millionen Dateien des Schweizerischen Bundesarchivs dokumentiert: 10-tap.sql.gz (fcb11b650d9c64a2f908a561934e6fd) und 10-log.tgz (d317855606603bca8033ab0ae21ea03e). Alle Dateien sind gzip-komprimiert.

### Verteilung TIFF-Komprimierung

In diesem Beispiel werden auf Grund der von *tiffhist* in die Logdatei geschriebenen Tag-Informationen die Verteilung und Werte des Compression Tags 259 untersucht. Das Tag 259 ist in der TIFF-Spezifikation folgendermassen erläutert:

**Compression**

Data can be stored either compressed or uncompressed.

Tag = 259 (103.H)
Type = SHORT
Value = 1 -10, values > 32766 (proprietary values)

| | |
|---|---|
| 1 | No compression |
| 2 | CCITT 1D |
| 3 | Group 3 Fax |
| 4 | Group 4 Fax |
| 5 | LZW |
| 6 | JPEG TIFF/6-.0 marked as deprecated |
| 7 | JPEG TIFF TechNote2 1995 |
| 8 | Adobe Deflate |
| 9 | JBIG bw |
| 10 | JBIG color |
| 32773 | PackBits |

**Abbildung 4: Definition des Compression Tag in der TIFF-Spezifikation**[9]

Ein einfaches Windows- oder Linux-Shellscript erzeugt aus den Logdateien die Übersicht in Tabelle 5.

9 TIFF Revision 6.0 (1992), siehe oben Anm. 4, S. 17.

| Compression | Basel-Stadt | St. Gallen | Bundesarchiv | Bundesarchiv extern | Total | Prozent |
|---|---|---|---|---|---|---|
| none | 732'696 | 550'675 | 560'956 | 217'843 | 2'062'170 | 52% |
| CCIT 1D | 0 | 0 | 564 | 0 | 564 | 0% |
| Fax Group 3 | 0 | 0 | 20'041 | 0 | 20'041 | 1% |
| Fax Group 4 | 22'745 | 317 | 602'571 | 1'207'376 | 1'833'009 | 46% |
| LZW | 31 | 15'651 | 19'534 | 0 | 35'216 | 1% |
| old JPEG | 0 | 0 | 0 | 0 | 0 | 0% |
| JPEG | 0 | 15'427 | 12'095 | 0 | 27'522 | 1% |
| Adobe Deflate | 0 | 0 | 1'593 | 0 | 1'593 | 0% |
| JBIG bw | 0 | 0 | 0 | 0 | 0 | 0% |
| JBIG color | 0 | 0 | 0 | 0 | 0 | 0% |
| Pack Bits | 0 | 0 | 0 | 0 | 0 | 0% |
| other | 0 | 0 | 0 | 0 | 0 | 0% |
| Total | 755'472 | 582'070 | 1'217'354 | 1'425'219 | 3'980'115 | 100% |

**Tabelle 5: Verteilung der Kompressionsarten im Korpus**

### Vergleich zweier Tools (exiftool und exiv2)

Die Auswertung der Ausgaben verschiedener Tools ist bei der Speicherung der Logausgabe in der Datenbank relativ einfach. Das SQL-Script vergleicht die Ausgabe von exiftool und exiv2 zur jeweils gleichen Datei und gibt das Resultat in einer HTML-Datei aus:

```
.output exiftool&exiv2.html
.mode asci
SELECT "<!DOCTYPE html><HTML><head><style> table { font-family: arial, sans-serif;
border-collapse: collapse; width: 100%; } td, th { border: 1px solid #dddddd; text-
align: left; padding: 8px; } tr:nth-child(even) { background-color: #dddddd; }
</style></head>
<BODY><PRE><TABLE>";

.mode html
SELECT
--  sys1.md5,
    sys1.toolname,
    sys1.sysout,
--  sys2.md5,
    sys2.toolname,
    sys2.sysout
FROM
    (SELECT md5, toolname, sysout from sysindex WHERE toolname = "exif") sys1
INNER JOIN
    (SELECT md5, toolname, sysout from sysindex WHERE toolname = "exiv2") sys2
ON
    sys1.md5 = sys2.md5;

.mode ascii
SELECT "</TABLE></PRE></BODY></HTML>";
.exit
```

**Abbildung 5: SQL-Script zum Toolvergleich**

Das Resultat, die Darstellung der Ausgabedatei in einem Browser, zeigt Abbildung 6.

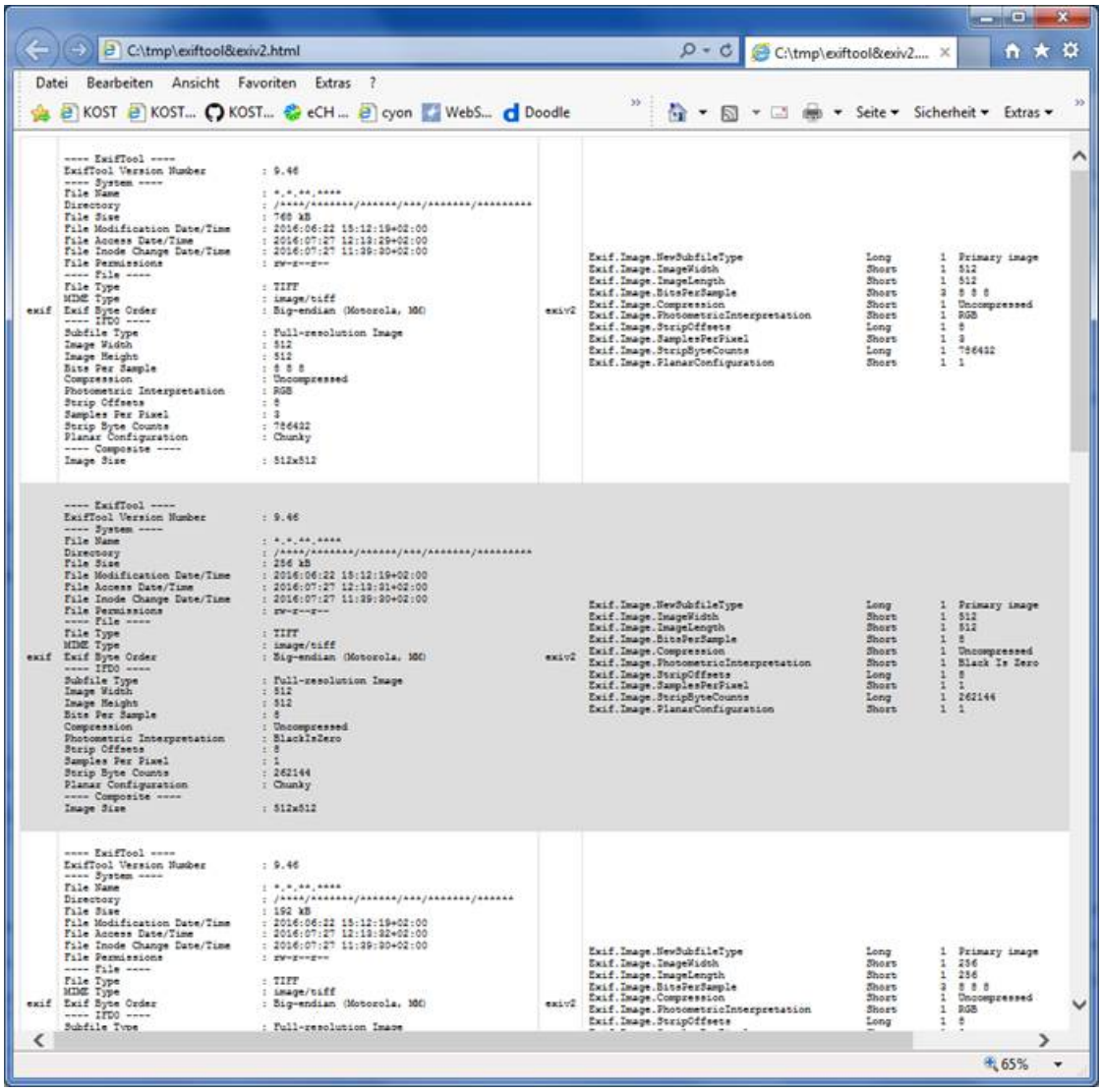


**Abbildung 6: Resultate des Toolvergleichs**

## Fazit

Die TIFF-Korpus-Analyse der KOST ermöglicht es der Fachgemeinschaft, Empfehlungen und Strategien zum Umgang mit TIFF-Dateien im Archiv auf konkrete Eigenschaften real existierender Dateien aus verschiedenen Archiven und Entstehungskontexten abzustützen. Die Publikation der Analysewerkzeuge und des Vorgehens erlaubt es zudem, weitere Korpora in vergleichbarer Weise zu analysieren und die Datenbasis damit zu vergrössern.