



Java Image Processing

Survival Guide

*“Siegfried, how difficult is it to
replace ImageMagick with a Java
image processing library?!”*



“Well, not too difficult”

Bilderansicht: Architektur in schönster Form!

www.willhaben.at/iad/viewimage?adId=95254947

WILLHABEN.AT®

HOL DIR DIE APP FÜR PHONE UND TABLET!

Speichern, teilen und veröffentlichen Sie Anzeigen auf willhaben.at

Login

Mein Willhaben Anzeigen Merkliste Suchagent

Immobilien(76758) Auto & Motor(115831) Jobs & Karriere(6040) Marktplatz(2546322)

Haus kaufen Detailsuche Suchergebnis Anzeige

Bilderansicht: Architektur in schönster Form!

Zurück zur Anzeige < Voriges Bild Fotoshow Nächstes Bild >

Top13_Wohnzimmer_v14

1 2 Nächste >







GREAT PLACE TO WORK Beste Arbeitgeber 2014 Österreich

Kontakt | will Werbung | Jobs bei willhaben.at | Presse | Mobile

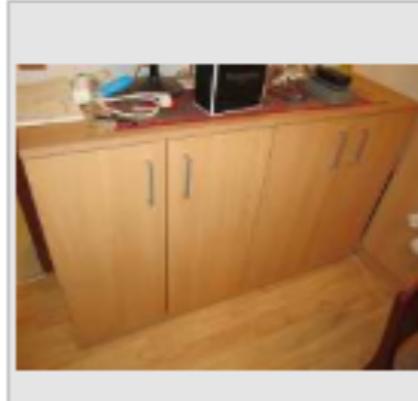
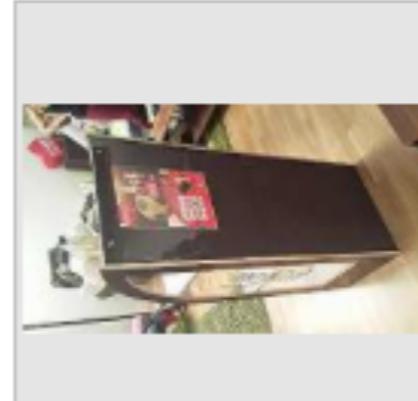
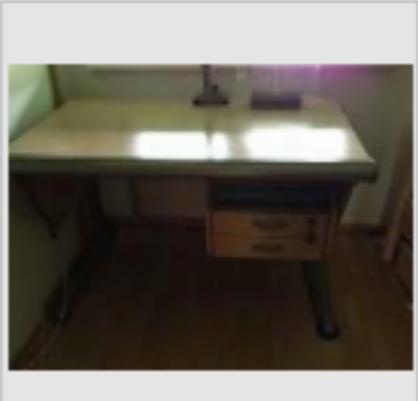
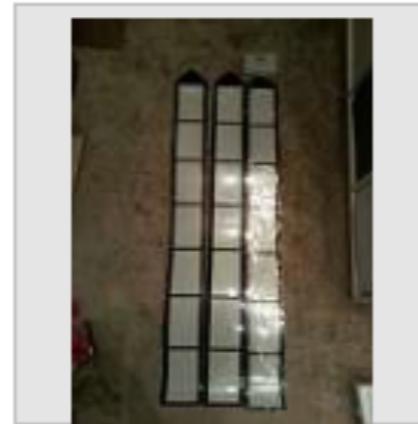
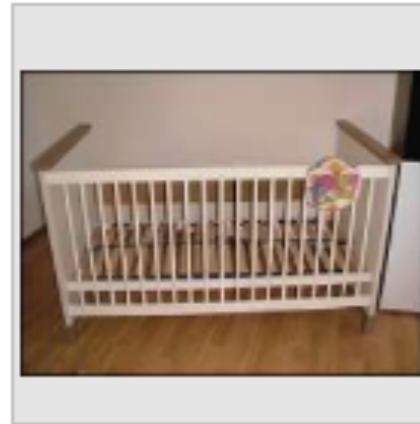
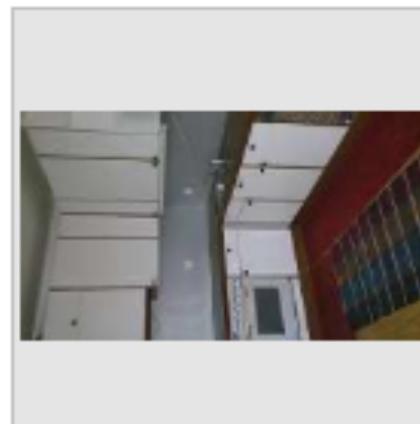
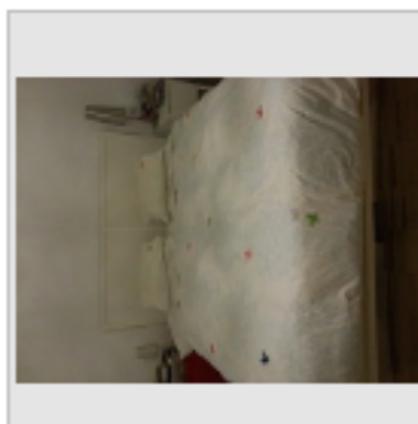
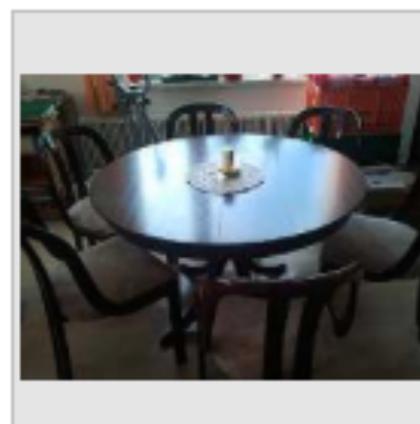
Hilfe | Sicherheitshinweise | Nutzungsbedingungen | AGB | Impressum | Sitemap

© 2014 willhaben. Alle Rechte vorbehalten.



218.189 Treffer

Veröffentlicht



The Customer's Approach

- ImageMagick native libraries
 - ▶ Handles image conversion and scaling
- Ghostscript for PDF Rendering
 - ▶ Invoked by ImageMagick
- JMagick to access ImageMagick libraries over Java Native Interface (JNI)

The Customer's Problems

- ImageMagick exceptions kills the JVM
 - ▶ Load balancer propagates the problem
- JMagick is no longer maintained
 - ▶ Unable to set JPEG quality parameter
- Installing ImageMagick, JMagick, Ghostscript and friends can be tricky
 - ▶ Failed miserably for Mac OS X

The Road Ahead

- Java ImageIO for low-level plumbing
- Java Advanced Imaging (JAI) for additional optional TIFF support
- Image scaling library to create previews and thumbnail images efficiently
- Apache PDFBox to create PDF previews

Java ImageIO

- Provides a *Service Provider Interface (SPI)*
 - ▶ Adding additional image formats
- *ImageReader* to create a *BufferedImage* from various image formats
 - ▶ JPEG, PNG, GIF & BMP
- *ImageWriter* to write the *BufferedImage*

Java ImageIO

- Apply one or more buffered image operations between reading and writing
 - ▶ Color conversion - grey-scaling
 - ▶ Affine transformation - scaling
 - ▶ Convolution operation - sharpening

Java Advanced Imaging

- Adds TIFF and JPEG2000 support
- Consists of three libraries
 - ▶ `jai_imageio.jar` provides SPI integration
- Libraries are not on Maven Central
- Last stable release in 2006
- Many dead documentation links

Image Scaling

- Many options when using Java 2D API
 - ▶ `Image.getScaledInstance()`
 - ▶ `Graphics.drawImage()`
 - ▶ `Graphics2D.drawImage()`
 - ▶ `BufferedImageOps`
 - ▶ `AffineTransforms`
- Better use dedicated image scaling library

Image Scaling

- Open Source image scaling libraries
 - ▶ *imgscalr*
 - ▶ *thumbnailator*
 - ▶ *java-image-scaling*
- Libraries have similar options, performance and result image quality

Image Scaling Quality

- JPEG quality settings
- Scaling algorithm being used
 - ▶ Nearest Neighbour, Bilinear, Bicubic Interpolation, Lanczos Re-sampling, ...
- Image optimisations
 - ▶ Sharpening, Anti-aliasing, Auto-Correction

Apache PDF Box

- Document Creation
- Text extraction
- PDF document manipulation
 - ▶ Merging & splitting
 - ▶ PDF to Image conversion
- PDF to Image conversion uses ImageIO SPI under the hood

Apache PDF Box

```
List<BufferedImage> pdfToImage(
    Object source, int from,
    int to, int dpi)
{
    PDDocument pdDocument = loadPDDocument(source);
    List<BufferedImage> result = new ArrayList<BufferedImage>();
    List<PDPage> pages = pdDocument.getDocumentCatalog().getAllPages();

    for (int i = from - 1; i < to && i < pages.size(); i++)
    {
        PDPage page = pages.get(i);
        BufferedImage image = page.convertToImage(BufferedImage.TYPE_INT_RGB, dpi);
        result.add(image);
    }
    return result;
}
```

Hitting
Real-life
Test Data



What's Wrong?!



PNG



JPEG

Alpha-Channels



- Alpha channel stores transparency for GIF and PNG
- ImageIO uses mismatched color model when saving as JPEG
- Resulting images have a red tint or are all black
- Need to convert to RGB using `Graphics2D.drawImage()`

Alpha-Channels

```
BufferedImage bufferedImage = createBufferedImage(sourceImageFile);
int width = bufferedImage.getWidth();
int height = bufferedImage.getHeight();
final int imageType = BufferedImage.TYPE_INT_RGB;
BufferedImage rgbBufferedImage = new BufferedImage(width, height, imageType);
Graphics2D graphics = rgbBufferedImage.createGraphics();
graphics.drawImage(bufferedImage, 0, 0, null);
graphics.dispose();
```

What's Wrong?!

```
if(imageSourceFile.length() > IMAGE_FILE_SIZE_LIMIT) {  
    throw new ImageConversionException("The image is too big ...");  
}
```

File Size versus Image Size

- Checking file size leads to *Decompression Bomb Vulnerability*
 - ▶ Images are usually compressed
 - ▶ 19.000×19.000 uni-color PNG uses 44 KByte disk space but 1 GB memory
- Use image metadata for width and height
- Better rescale images before uploading

What's Wrong?!

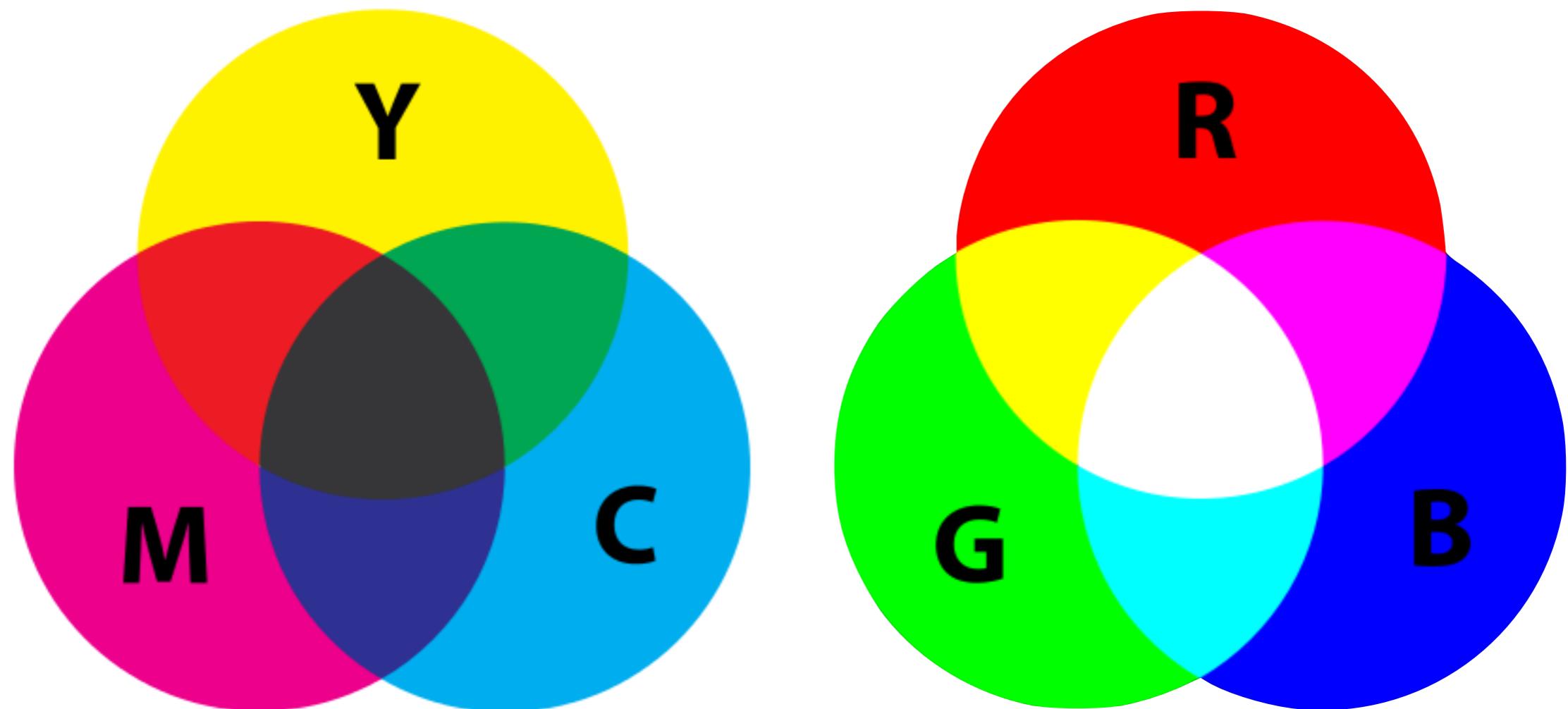


JPEG

```
javax.imageio.IIOException: Unsupported Image Type
```

```
at com.sun.imageio.plugins.jpeg.JPEGImageReader.readInternal(JPEGImageReader.java:1063)
at com.sun.imageio.plugins.jpeg.JPEGImageReader.read(JPEGImageReader.java:1034)
at javax.imageio.ImageIO.read(ImageIO.java:1448)
at javax.imageio.ImageIO.read(ImageIO.java:1308)
```

CMYK Color Model



CMYK Color Model

- CMYK is a subtractive color model
- Color printing and hence Photoshop users
- Java ImageIO JPEGImageReader will not read CMYK images
- In the need of TwelveMonkeys library



12 MONKEYS

Source: <https://fanart.tv/movie/63/12-monkeys/>

TwelveMonkeys History

- Named after the movie and a long night in a pub with colleagues
- Initially used for a web-based CMS and later for an Online Bookstore
 - ▶ Get the work done
 - ▶ Make it easy to work with images
 - ▶ Learn about Color Theory, ICC profiles, compression algorithms, ...

TwelveMonkeys History

- Initially reading Photoshop images
 - ▶ PSDImageReader
- Solving JPEG production issues
 - ▶ CMYK color space
 - ▶ AdobeRGB ICC profile support
 - ▶ Broken ICC color profiles handling
 - ▶ Inconsistent JPEG metadata

ImageIO & Web Apps

- ImageIO Plugin Registry is VM global and doesn't play nicely with web apps
- Need to invoke `ImageIO.scanForPlugins()` to discover plugins over SPI
 - ▶ Class-loader issue & resource leaks
- TwelveMonkeys `IIOProviderContextListener`
 - ▶ Dynamic loading/unloading of plugins

ImageIO & Web Apps

```
<web-app>
  <listener>
    <display-name>
      ImageIO service provider loader/unloader
    </display-name>
    <listener-class>
      com.twelvemonkeys.servlet.image.IIOProviderContextListener
    </listener-class>
  </listener>
</web-app>
```

Why Use TwelveMonkeys?

- Uses standard ImageIO API
- Actively maintained
- Large and growing user base
- No native code
- Open source with liberal license (BSD)

TwelveMonkeys Road Map

- Improving JPEG support
 - ▶ Lossless JPEG
 - ▶ Arithmetic coding
 - ▶ CMYK write support

TwelveMonkeys Road Map

- Camera RAW plugins
 - ▶ CR2 (Canon)
 - ▶ NEF (Nikon)
 - ▶ TIFF/EP and DNG (ISO/Adobe)

TwelveMonkeys Road Map

- Improving TIFF support
 - ▶ Read support for CCITT T4 and T6
 - ▶ Write support
 - ▶ LZW & all baseline compressions
 - ▶ JPEG (Compression Type 7)
 - ▶ Tiles/Strips
 - ▶ Metadata



In Production

Source : www.wikipedia.org

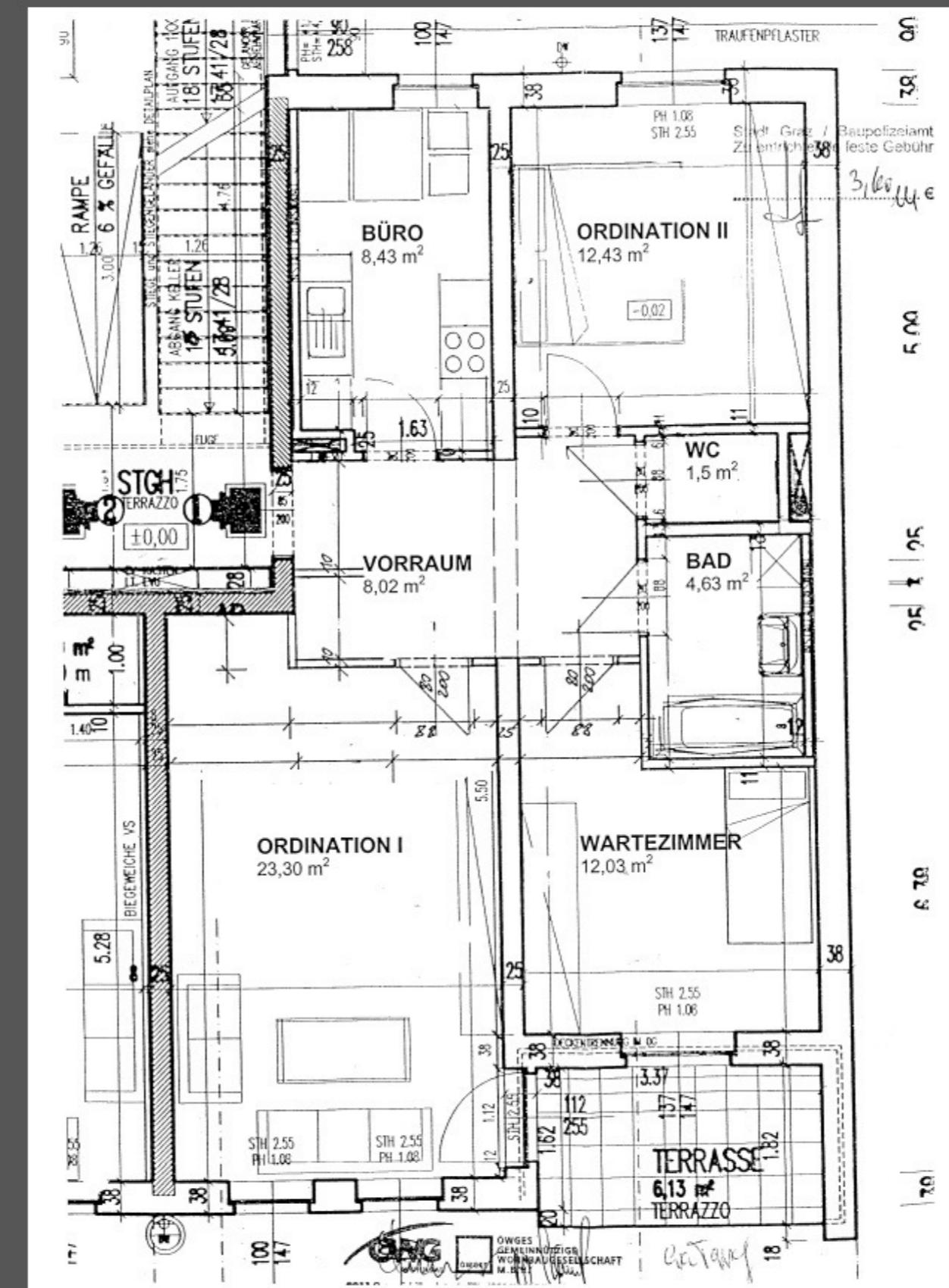
ImageIO Segment Violation

```
#  
# A fatal error has been detected by the Java Runtime Environment:  
#  
# SIGSEGV (0xb) at pc=0x00007fce89c42353, pid=20878, tid=140525025629952  
#  
# JRE version: Java(TM) SE Runtime Environment (7.0_45-b18) (build 1.7.0_45-b18)  
# Java VM: Java HotSpot(TM) 64-Bit Server VM (24.45-b08 mixed mode linux-amd64 compressed o  
ops)  
# Problematic frame:  
# C  IntArgbPreToIntRgbSrcOverMaskBlit+0x2a3  
#  
Stack: [0x00007fce882c1000,0x00007fce883c2000],  sp=0x00007fce883b8428,  free space=989k  
Native frames: (J=compiled Java code, j=interpreted, Vv=VM code, C=native code)  
C  [libawt.so+0x77353]  IntArgbPreToIntRgbSrcOverMaskBlit+0x2a3  
C  [libawt.so+0x45017]  Java_sun_java2d_loops_TransformHelper_Transform+0xeb7  
j  sun.java2d.loops.TransformHelper.Transform(Lsun/java2d/loops/MaskBlit;Lsun/java2d/Surfac  
eData;Lsun/java2d/SurfaceData;Ljava.awt/Composite;Lsun/java2d/pipe/Region;Ljava/awt/geom/Af  
fineTransform;IIIIIIII[III)V+0  
j  sun.java2d.pipe.DrawImage.renderImageXform(Lsun/java2d/SunGraphics2D;Ljava/awt/Image;Ljava  
/awt/geom/AffineTransform;IIIIILjava/awt/Color;)V+505  
j  sun.java2d.pipe.DrawImage.transformImage(Lsun/java2d/SunGraphics2D;Ljava/awt/Image;IILja  
va/awt/geom/AffineTransform;I)V+366  
j  sun.java2d.pipe.DrawImage.transformImage(Lsun/java2d/SunGraphics2D;Ljava/awt/Image;Ljava  
/awt/geom/AffineTransform;Ljava/awt/image/ImageObserver;)Z+17  
j  sun.java2d.pipe.ValidatePipe.transformImage(Lsun/java2d/SunGraphics2D;Ljava/awt/Image;Ljava  
/awt/geom/AffineTransform;Ljava/awt/image/ImageObserver;)Z+17  
j  sun.java2d.SunGraphics2D.drawImage(Ljava/awt/Image;Ljava/awt/geom/AffineTransform;Ljava/  
awt/image/ImageObserver;)Z+111  
j  org.apache.pdfbox.pdfviewer.PageDrawer.drawImage(Ljava/awt/Image;Ljava/awt/geom/AffineTr  
ansform;)V+35
```

Page Thumbnails



1



ImageIO Segment Violation

- PDF with 10200 x 13992 pixel image scan
 - ▶ 330 KByte file size
- Native ImageIO code bombs and kills JVM
- Load balancer propagates the problem
- Determine the size of the embedded image and reject if it exceeds the limits

ImageIO Segment Violation

```
PDPage page = ...  
for (PDXObject embeddedObject : page.getResources().getXObjects().values()) {  
    if (embeddedObject instanceof PDXObjectImage) {  
        int width = ((PDXObjectImage) embeddedObject).getWidth();  
        int height = ((PDXObjectImage) embeddedObject).getHeight();  
    }  
}
```

**Thousands
Incompatible
File
Formats**

TIFF

- Tagged Image File Format
- Supports dozen of image, storage and compression formats
- It is practically impossible to correctly process all real-life TIFF image files
- Baseline TIFF as minimal functionality
- But “TIFF 6.0, Part 2:TIFF Extensions” are commonly used in real-life

Baseline TIFF

- Multi-page support
- Image storage as strips or tiles
- Image Types - bilevel, grayscale, palette-color and RGB full-color images
- Compression schemes - uncompressed, CCITT Group 3 encoding, PackBits
- Little and Big Endian byte order
- Handling of optional fields

TIFF 6.0 Extensions

- Additional compression schemes - CCITT T.4 & CCITT T.6 bi-level encoding, LZW, JPEG-based compression
- Additional image types - CMYK, YCbCr, HalftoneHints, Tiled Images, CIE L*a*b*
- Tiff 6.0 Extension also supported by JAI & Apache Commons Imaging
 - ▶ Glue code uses fallback to different TIFF readers to handle parsing errors

What's Wrong?!

VOLVO

TRANSPORTMELDING

Nr. [REDACTED] COPY

Van: [REDACTED]
Bestemming: [REDACTED]
Tel.: [REDACTED]
Fax: [REDACTED]

Uitgever: [REDACTED]
Afdeling: [REDACTED]

Transportfirma: [REDACTED] TM - datum - uur: [REDACTED]

Laad eenheid: [REDACTED] ETA: [REDACTED]

Type: TRA Transportwijze: ROAD

Opmerkingen:

Vagnr. Lossing	Losplaats	Unit	Bestemming Suffix	Poort	Info	Prior	Tpt. Colli	Bruto gewicht	Volume (kub.meter)
1	VET	VET	07A				13	1389,99	11,674
1	VET	VET	070				6	574,00	5,229
2	KNVET	KNVET	07B				56	7000,00	54,320

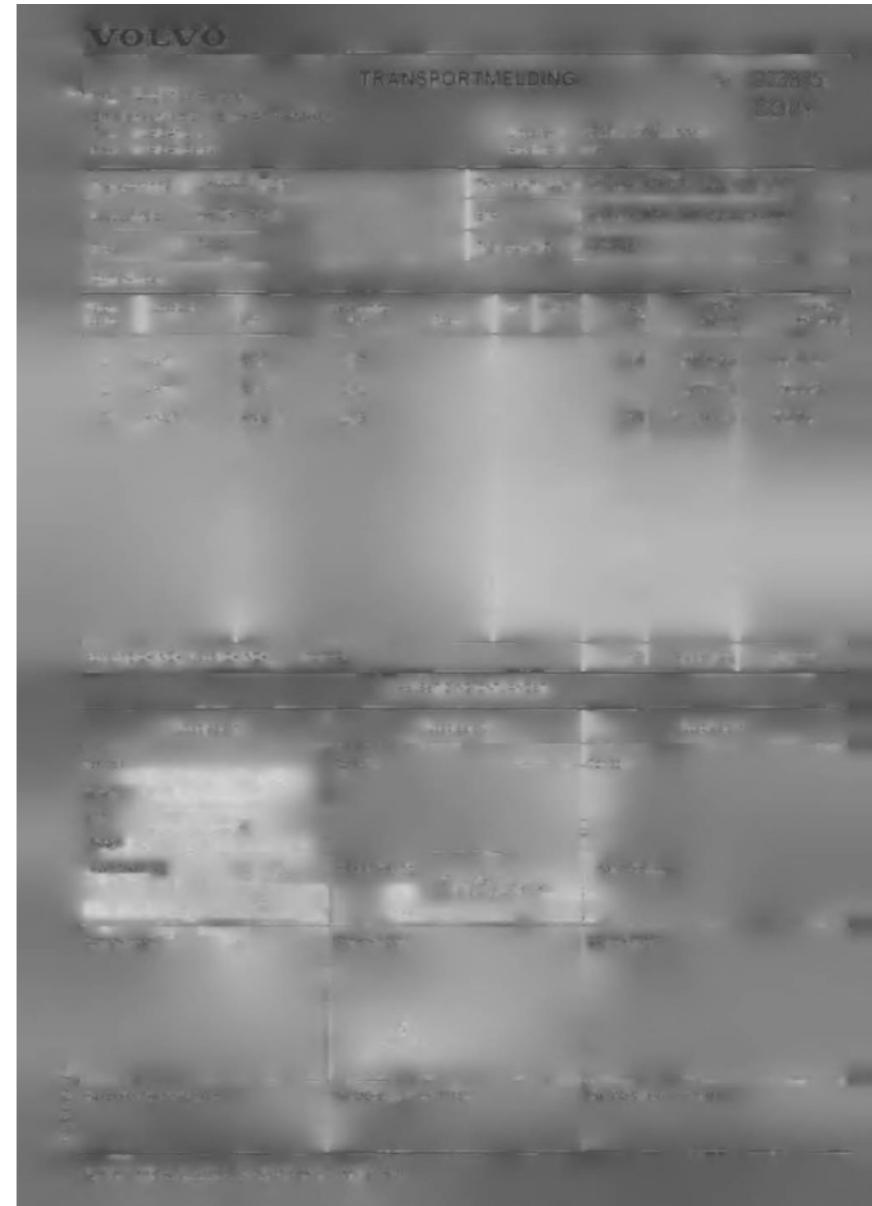
Sortering OK/NOK Label OK/NOK TOTAAL: 75 8963,99 71,223

BEVESTIGING ONTVANGST

Losplaats 1	Losplaats 2	Losplaats 3
Tpt colli	Tpt colli	Tpt colli
Datum	Datum	Datum
Unit	Unit	Unit
Naam	Naam	Naam
Handtekening	Handtekening	Handtekening
Opmerkingen: Handtekening		
Handtekening vervoerder	Handtekening vervoerder	Handtekening vervoerder

F4.9.D-F01 HERZ.2 , 2001.06.01VERG. VERV. COM.: 2695001

PDF



JPEG

PDFBox & JBIG2

- PDF contains a JBIG2 image
- JBIG2 was added to PDF 1.4 in 2004
- PDFBox relies on ImageIO to convert the source image into a JPEG
- No suitable ImageIO plugin was found and PDFBox returned a JPEG nevertheless

PDFBox & JBIG2

ERROR JBIG2Filter - Can't find an ImageIO plugin to decode the JBIG2 encoded datastream.

ERROR PDPPixelMap - Something went wrong ... the pixelmap doesn't contain any data.

WARN PDXObjectImage - masking getRGBImage returned NULL

ImageIO & JDK 8

- Kodak CMS (Color Management System)
replaced by Little CMS
- Widely used, maintained & open source
- Some ICC profiles will crash/break on
reading
- For now (Java 8) it's possible to revert to
the “old” KCMS using the switch
 - ▶ `-Dsun.java2d.cmm=sun.java2d.cmm.kcms.KcmsServiceProvider`

A photograph of three students sleeping in a lecture hall. In the foreground, a woman with dark hair, wearing a white shirt and a gold watch, rests her head on her hands on a wooden desk. In the middle ground, a man in a dark suit jacket and white shirt is also asleep with his head down. In the background, another person in a red jacket is visible, also sleeping. The room has rows of wooden desks and chairs.

Things to take home

Things To Take Home

- You won't handle all of your real-life images & PDFs without issues
 - ▶ Get as many test documents as possible
 - ▶ Be prepared to hit the learning curve
- Stay clear of TIFF if possible
- Never use the term “TIFF” - use “Baseline TIFF” or “Baseline TIFF & TIFF 6.0” instead

Things To Take Home

- `TwelveMonkeys` bells & whistles
 - ▶ Scaling, color conversion, cropping, water-marking
 - ▶ Real-life JPEG support
- JAI and/or Apache Imaging for additional TIFF support
- Check out dedicated image scaling libraries if performance is critical

[https://github.com/sgoeschl/
java-image-processing-
survival-guide](https://github.com/sgoeschl/java-image-processing-survival-guide)

Questions & Answers



Unused Stuff

Common Image Size

Image Source	Resolution	Megapixel	MB
iPhone 5	3.264 x 2.448	8	22
A4 page 300 DPI	3.437 x 2.480	8	22
Nikon D610	6.016 x 4.016	24	68
A4 page 600 DPI	6.874 x 4.960	32	96
Nokia Lumia 1020		38	114

**“Siegfried, how difficult is it to replace
our Image Optimisation Server”**



“Well, that will be difficult”