

Aufgabe 3: Bildverarbeitung in einer komponentenbasierten Architektur (6 Pkte)

Entwickeln Sie die Pipes&Filter Architektur zur Bildverarbeitung aus Aufgabe 2 weiter zu einer komponentenbasierten Java-Beans Architektur.. Es sollen die Filter, Source und Sink wie in Aufgabe 2 nun als konfigurierbare Komponenten in der BeanBox angeboten werden.

Es sollen folgende Komponenten für die Java Beanbox entwickelt werden auf Basis von JAI oder Catalano:

- ~~- Bild laden aus einer Datei~~
- ~~- Bild speichern in einer Datei~~
- ~~- Bild anzeigen~~
- ~~- Dilate~~ i
- ~~- Erode~~ du
- Opening i
- Closing du
- Median i
- Threshold du
- ~~- ROI (Koordinaten des Bildausschnittes als Properties eingeben.~~ Optional - ein Extrapunkt: Benutzer kann interaktiv ein Rechteck über die region-of-interest des Bildes mit der Maus ziehen, wie in Beschreibung von Aufgabe 2 angegeben)
- Bildüberlagerung: eine Komponente mit zwei Eingängen, die in ihrem Ausgang ein Bild herausgibt das durch Überlagerung zweier Einzelbilder entsteht: wir wollen dabei vor allem weisse „Bond Balls“ auf den Lötstellen des Originalbildes sehen.

Die Komponenten sollen dem JavaBeans Komponentenmodell folgen. Insbesondere sollen sie im Beanbox Container per Mausinteraktionen zusammengesetzt werden können in einer Pipes&Filter Struktur: Komponenten werden in den Container gebracht (drag und drop) und über einen geeigneten Event-Mechanismus verbunden.

Folgende Anforderungen sind insbesondere zu beachten:

1. Die Eigenschaften der Komponenten, die im Eigenschaftseditor erscheinen und die der Anwender editieren möchte zur Anpassung der Komponenten an das spezielle Bildverarbeitungsproblem, sollen nur aus dem Anwendungsbereich (Bildverarbeitung) stammen: BeanInfo Klassen benutzen.
2. Masken sollen bequem (!) spezifizierbar sein, aber ohne grossen Aufwand in der GUI: der Property-Editor der Beanbox sollte reichen (d.h. kein Customizer)
3. Bei Änderung einer Property irgendeines Operators soll die Pipeline **automatisch** Resultatbilder bzw. das Anwendungsergebnis (Anzahl Kastanien) neu berechnen: der Benutzer will ja gerade mit der Beanbox die Anwendung (Pipeline) auf die richtigen Properties interaktiv einstellen, damit das erwünschte Resultat erreicht wird.

Abgabe:

was? die Komponenten im richtigen Deploymentformat; den Sourcecode; Designdokumentation (UML: Klassendiagramm); Beschreibung (kleines Manual), welche Properties man wie für die einzelnen Beans eingibt.