Handout Python-Einführung

KOGS - Module Konzept

Wenn man einen Account bei KOGS bekommt, muß man sich zuerst daran gewöhnen, daß man Software in der Regel nur über die module-Umgebung erreichen kann. D.h. wenn man die entsprechenden (Software-)Module nicht lädt, hat man die meiste Software entweder gar nicht, oder nur in älterer (bis uralter!) Version zur Verfügung.

Die wichtigsten Befehle sind

module avail - zeigt die verfügbaren Module an
module add foo [bar baz ...] - lädt das Modul foo (ggf. auch bar und baz im gleichen Durchlauf)
module help foo - zeigt Informationen zum Modul foo an

Meine Standard-Umgebung unter Solaris enthält 33 Module, sowas schreibt man sich aber dann natürlich dann in die .bashrc bzw. .cshrc. Am wichtigsten ist sicherlich "module add vigra", aber unter Solaris finde ich auch ein module add gnu-utils bzip2 gcc unausweichlich.

Start von VIGRA Interactive

module add vigra
vigra

Erste Schritte in Python

Tutorial: http://docs.python.org/tut/
Library Dokumentation: http://docs.python.org/lib/

Grundoperationen im VIGRA Terminal

(alle auch über das Menü "Vigra" erreichbar)

```
img = readImage("/path/to/image.jpg") # kann natürlich auch andere Formate (PNG, TIFF, ...)
img = GrayImage(100, 200)             # erzeugt ein "leeres" (schwarzes) Bild (analog: RGBImage)

win = showImage(img)                  # Bild anzeigen (Anzeige wird nicht automatisch aktualisiert)

pos = (10, 10)                # Positionen / Größen können immer als 2-Tupel/Paar angegeben werden
img[pos] = 100                # Lese-/Schreibzugriff auf einzelne Pixel
img[20,10] = (0, 0, 50)       # dito., bei RGB-Bild (Beispiel: sehr dunkles Blau)

img.write("test.png", BYTE)   # Bild speichern, Pixelwerte = 0-255 (schwarz..weiß)
img.write("look.png", NBYTE)  # normalisiertes Bild speichern, vorhandener Wertebereich wird auf
                              # schwarz..weiß abgebildet (z.B. auch 0.0 bis 1.0 oder -12 bis 23.4)

# Schleife über alle Pixel:
for y in range(img.height()):
    for x in range(img.width()):
        img[x, y] = 255-img[x, y]

# Beispiel X-Differenz (einfachster horizontaler Kantenfinder)
for y in range(img.height()):
    for x in range(img.width()-1):
        img[x+1, y] -= img[x, y] # Pixelwert von links abziehen
```

Python Tutorial unter http://www.python.org/doc/current/tut/