Modeling a functional curve in a digital image

⊞— Image

⊟

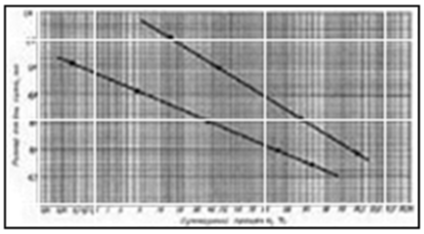
Curve image in gray scale: $M := Mean(\text{image2rgb}(\text{"img.gif"}))$

Validation of the data computing the average value of each row and column, then compares this value with kline which has represents the gray value selected to define the maximum "black" pixel.

$$clean(M\#, k_{11}, k_{12}) := \begin{array}{|l} [\, M := M\#\ \ Mc := M\ \ R := 0\ \ C := 0 \,] \\ [\, m := [1..\text{rows}(M)]\ \ n := [1..\text{cols}(M)] \,] \\ [\, R_m := Mean(\text{row}(M, m))\ \ C_n := Mean(\text{col}(M, n)) \,] \\ [\, R\ \ C \,] := \text{eval}\left(\left[\ \overrightarrow{k_{11} \le R \le k_{12}}\ \ \overrightarrow{k_{11} \le C \le k_{12}}\ \right]\right) \\ \text{for } m \in [1..\text{rows}(M)] \\ \quad \text{for } n \in [1..\text{cols}(M)] \\ \quad\quad \text{if } R_m \wedge C_n \\ \quad\quad\quad \textbf{continue} \\ \quad\quad \text{else} \\ \quad\quad\quad Mc_{m\,n} := 255 \\ Mc \end{array}$$

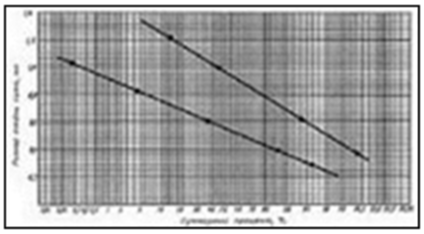
$k_1 := 130$ $\quad Mc := clean(M, k_1, 255)$ $\quad$ Try for different kl values until find a good one.

Cleaned data



Mc

Original data



M

With the matrix cleaned we can look for pixels from the function curve. With the following function we can scan the cleaned image matrix in the vertical and horizontal directions to find pixels that belong to the curve; the maximum opaque color is contained in the variable kcurve.

$$scan(M, k_{c1}, k_{c2}) := \begin{array}{|l} [\, nc := \text{cols}(M)\ \ n := [1..\text{length}(M)]\ \ A := 0 \,] \\ v := \text{col}\left(\text{findrows}\left(\text{eval}\left(\text{augment}\left(\overrightarrow{k_{c1} \le M \le k_{c2}}_n, n\right)\right), 1, 1\right), 2\right) \\ [\, r\ \ c \,] := \text{eval}\left(\left[\ \overrightarrow{1 + \text{trunc}\left(\frac{v-1}{nc}\right)}\ \ \overrightarrow{1 + \text{mod}(v-1, nc)}\ \right]\right) \\ \text{eval}(\text{csort}(\text{augment}(c, \text{rows}(M) - r + 1), 1)) \end{array}$$

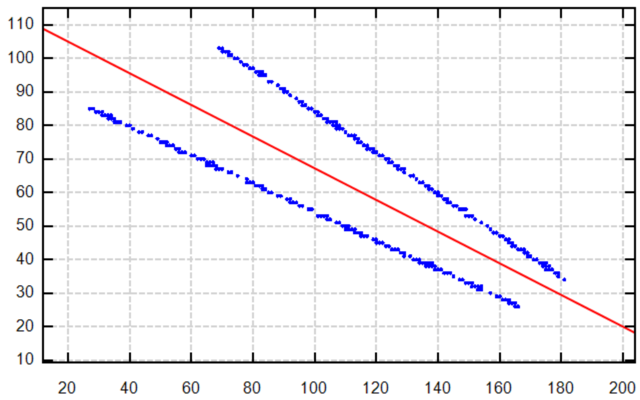
$k_c := 80$ $\quad Ms := scan(Mc, 0, k_c)$ $\quad$ Try for different kc values until find a good one.

Also isolate the curves selectring adequate ranges

Read range values here $\quad xr := [20..200]$ $\quad yr := [20..105]$

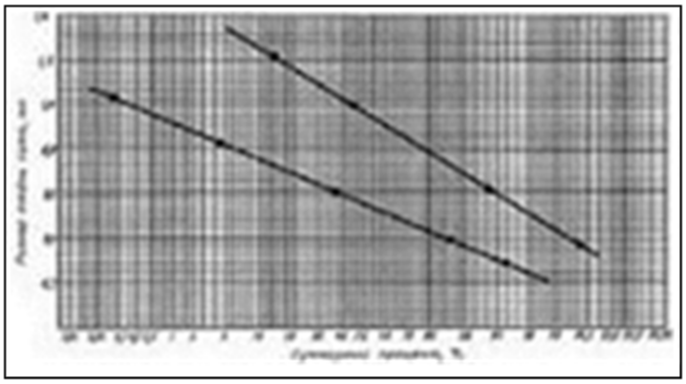
$$separe(x) := \frac{\max(yr) - \min(yr)}{\max(xr) - \min(xr)} \cdot (\max(xr) - x) + \min(yr)$$

Scanned data



augment(Ms, ".")
separe(x)

Original data



M

Now we can fit the data according to some model

$$\lambda(x, m, n) := \overrightarrow{m \cdot x + n}$$

$A := \text{eval}(\text{matrix}(0, 2))$ $B := A$

```
for r ∈ [1..rows(Ms)]
  if separe(Ms[r 1]) < Ms[r 2]
    A := eval(stack(A, row(Ms, r)))
  else
    B := eval(stack(B, row(Ms, r)))
```

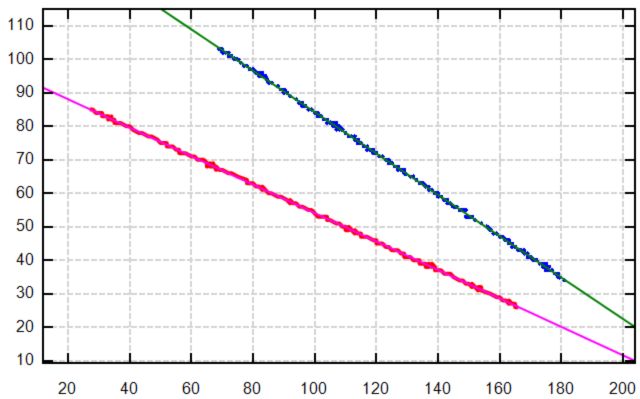
$$\lambda(u, M) := \lambda(\text{col}(M, 1), u_1, u_2) - \text{col}(M, 2)$$

$$\begin{bmatrix} mA \\ nA \end{bmatrix} := \text{al\_nleqsolve}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \lambda(u) := \lambda(u, A)\right)$$

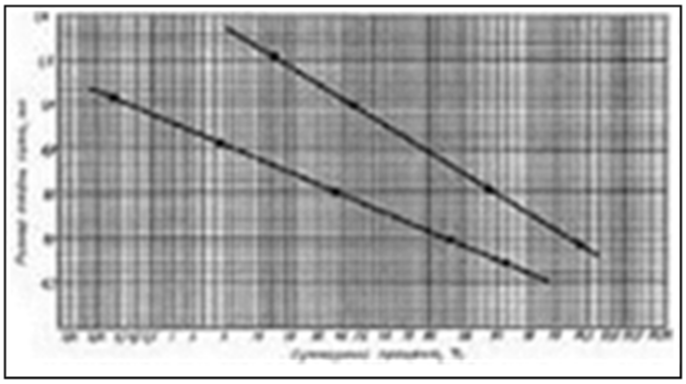
$$\begin{bmatrix} mB \\ nB \end{bmatrix} := \text{al\_nleqsolve}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \lambda(u) := \lambda(u, B)\right)$$

Results   ToDo: scale the plot

$$\begin{bmatrix} mA \\ nA \end{bmatrix} = \begin{bmatrix} -0.6174 \\ 145.9792 \end{bmatrix} \qquad \begin{bmatrix} mB \\ nB \end{bmatrix} = \begin{bmatrix} -0.4247 \\ 96.5775 \end{bmatrix}$$



A
B
λ(x, mA, nA)
λ(x, mB, nB)



M

Alvaro   appVersion(4) = "1.0.8348.30405"