Procesarea Imaginilor Digitale

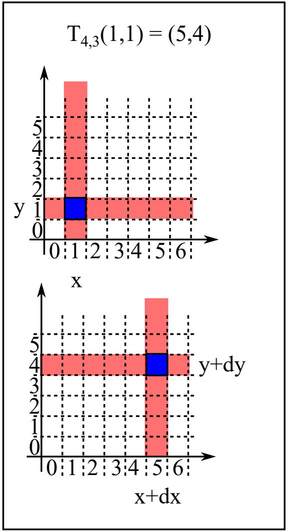
Curs - Transformări geometrice - Interpolare

Universitatea ”Transilvania” din Brașov

Transformări geometrice de bază

1. Translația.
2. Scalarea.
3. Rotația.

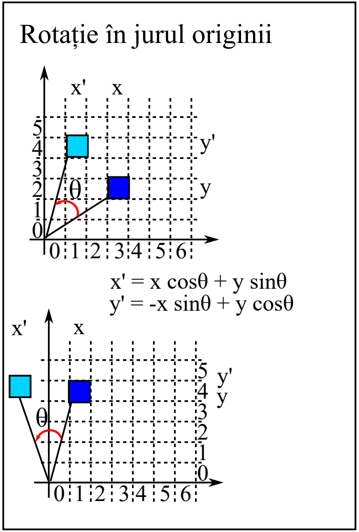
Transformări geometrice de bază



Transformări geometrice adupra unui punct

- **Translația**:
  $T_{dx,dy}(x, y) = (x + dx, y + dy)$

Transformări geometrice de bază

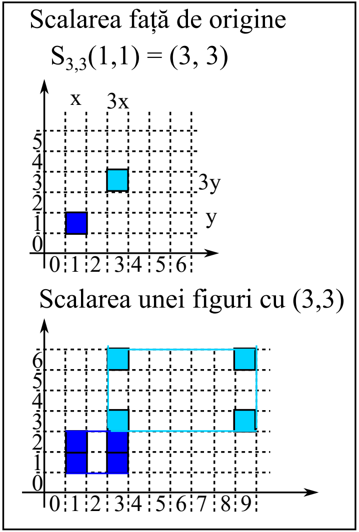


Transformări geometrice adupra unui punct

- **Translația**:
  $T_{dx,dy}(x, y) = (x + dx, y + dy)$

- **Rotația**:
  $$R_\theta(x, y) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$
  $$= (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$$

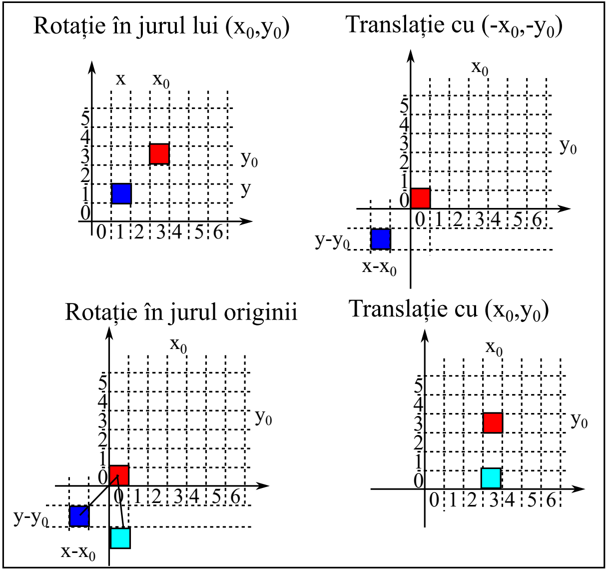
Transformări geometrice de bază



Transformări geometrice adupra unui punct

- **Translația**:
  $T_{dx,dy}(x,y) = (x+dx, y+dy)$
- **Rotația**:
  $$R_\theta(x,y) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$
  $$= (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$$
- **Scalarea**: $S_{sx,sy}(x,y) = (s_x x, s_y y)$

Transformări geometrice - Rotația în jurul unui punct dat



Imaginea rotită: în jurul unui punct oarecare (x_0, y_0):

$$g(x', y') = f(x, y),$$

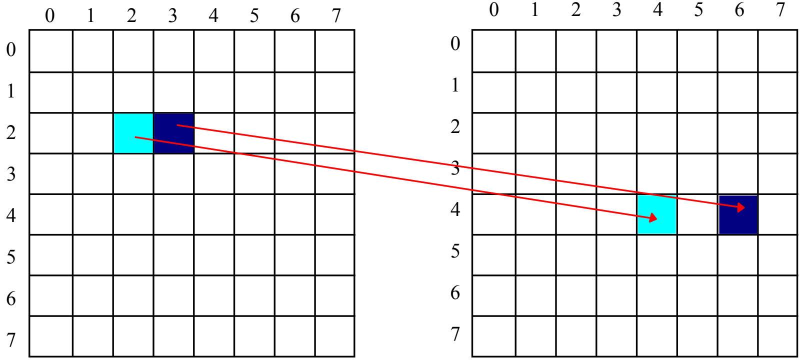
unde

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

Interpolarea - Considerente generale

- Aplicarea directă a formulelor $\Rightarrow$ "goluri" imaginea rezultat.
- Scalarea: informație numai pentru pixelii de pe poziții multiplii ai factorului de scalare.
- Rotația: pierdere de informație prin trunchierie,

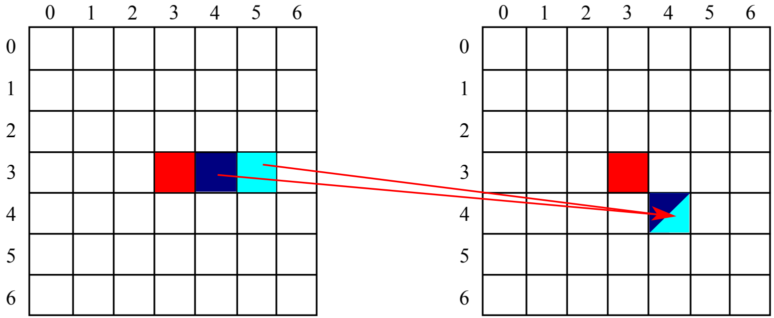
Scalarea imaginilor



Imagine sursă $(f) \longrightarrow$ imagine rezultat (g):

$$g(x', y') = g(ax, ay) = f(x, y)$$

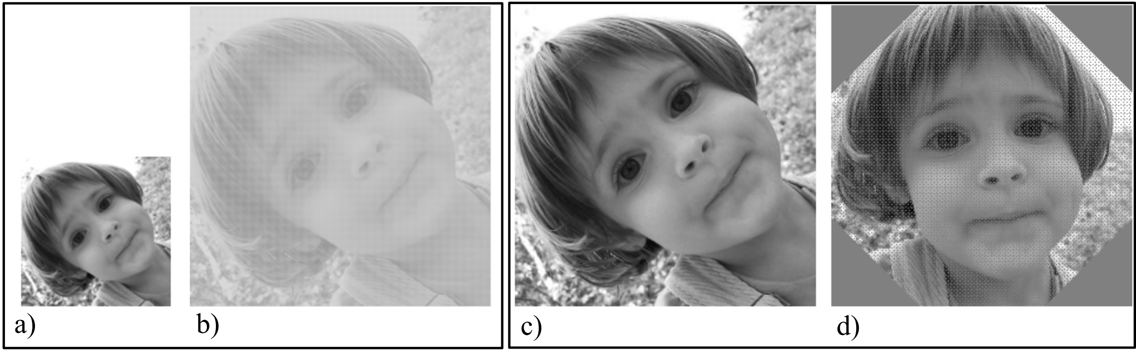
Rotația în jurul centrului imaginii



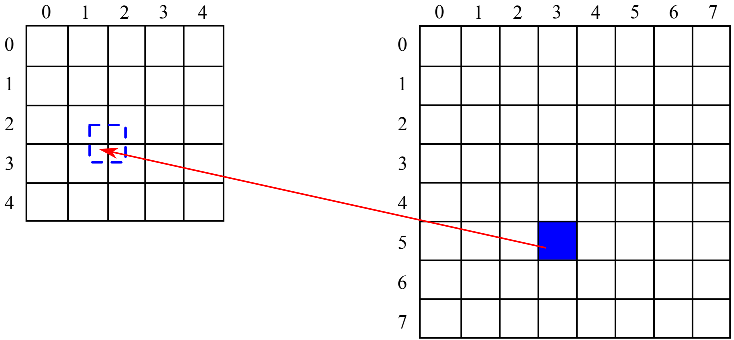
Imagine sursă $\longrightarrow$ imagine rezultat:

$$\begin{cases} x' = (x - x_0)\cos\theta - (y - y_0)\sin\theta + x_0 \\ y' = (x - x_0)\sin\theta + (y - y_0)\cos\theta + y_0 \end{cases}$$

Transformări geometrice - Aplicarea directă a formulelor

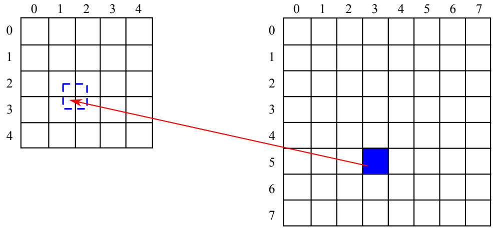


Scalarea imaginilor - De la imaginea rezultat către imaginea sursă



Imagine rezultat $\longrightarrow$ imagine sursă: ce valoare aleg?

Scalarea imaginilor - De la imaginea rezultat către imaginea sursă

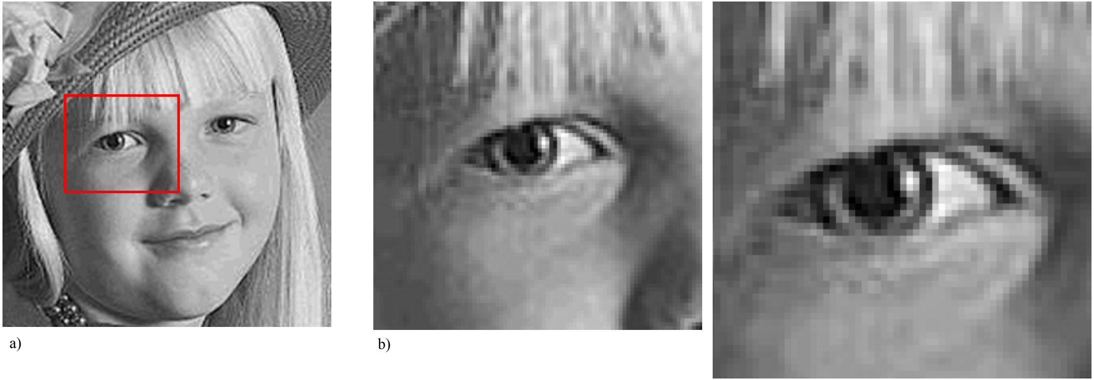


Aproximarea prin cel mai apropiat vecin.

Algoritm 1: Scalarea imaginii cu factorii sx, sy

pentru $y = 0$, *Rezult.height-1* **executa**
  pentru $x = 0$, *Rezult.width-1* **executa**
    $xc \leftarrow x/sx$ ▷ `valoare reală`
    $yc \leftarrow y/sy$ ▷ `valoare reală`
    $Result[y, x] = Input[round(yc), round(xc)]$ ▷ `rotunjesc coordonatele`
  sfarsit_for
sfarsit_for

Scalarea imaginilor de la rezultat către sursă - cel mai apropiat vecin



a) b) c)

Figura: a) Imaginea de test originală; b) Fragment din imaginea scalată cu factorul 3 în ambele direcții; c) Fragment din imaginea scalată cu factorul 5 în ambele direcții.

Interpolarea

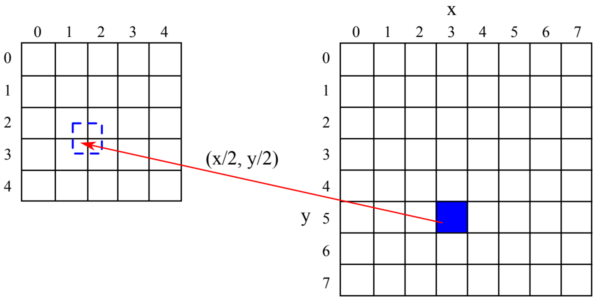
Problematică: Se cunosc

$$f(x_i) = y_i, i = \overline{0, n},$$

cu $x_0 < x_1 < \ldots < x_n$.

Să se aproximeze $f(x)$ $\forall x \in [x_0, x_n]$

Interpolarea - Pentru scalare



- pixelul (x, y) din imaginea rezultat "provine" din pixelul "real" $(x_c, y_c) = (\frac{x}{2}, \frac{y}{2})$
- cunoaștem valorile imaginii sursă în pixelii întregi: $([x_c], [y_c])$, $([x_c] + 1, [y_c])$, $([x_c], [y_c] + 1)$, $([x_c] + 1, [y_c] + 1)$

Care va fi valoarea pentru pixelul "real" (x_c, y_c)?

Tipuri de interpolare pentru imagini

- Interpolare prin cel mai apropiat vecin: rotunjesc coordonatele obținute la cele mai apropiate coordonate întregi $\Rightarrow$ se formează blocuri de culoare uniformă

Tipuri de interpolare pentru imagini

- Interpolare prin cel mai apropiat vecin: rotunjesc coordonatele obținute la cele mai apropiate coordonate întregi $\Rightarrow$ se formează blocuri de culoare uniformă
- Interpolare bilinară: se creează o nouă valoarea ca o combinație liniară a celor mai apropiați 4 vecini

Tipuri de interpolare pentru imagini

- Interpolare prin cel mai apropiat vecin: rotunjesc coordonatele obținute la cele mai apropiate coordonate întregi $\Rightarrow$ se formează blocuri de culoare uniformă
- Interpolare bilinară: se creează o nouă valoarea ca o combinație liniară a celor mai apropiați 4 vecini
- Interpolare bicubică: se creează o nouă valoare ca o combinație a 16 vecini (vom discuta)

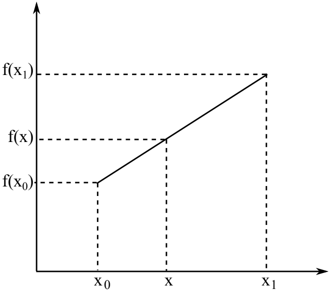
Tipuri de interpolare pentru imagini

- Interpolare prin cel mai apropiat vecin: rotunjesc coordonatele obținute la cele mai apropiate coordonate întregi $\Rightarrow$ se formează blocuri de culoare uniformă
- Interpolare bilinară: se creează o nouă valoarea ca o combinație liniară a celor mai apropiați 4 vecini
- Interpolare bicubică: se creează o nouă valoare ca o combinație a 16 vecini (vom discuta)
- Interpolare direcționată după contur - *Edge-directional interpolation*

Tipuri de interpolare pentru imagini

- Interpolare prin cel mai apropiat vecin: rotunjesc coordonatele obținute la cele mai apropiate coordonate întregi $\Rightarrow$ se formează blocuri de culoare uniformă
- Interpolare bilinară: se creează o nouă valoarea ca o combinație liniară a celor mai apropiați 4 vecini
- Interpolare bicubică: se creează o nouă valoare ca o combinație a 16 vecini (vom discuta)
- Interpolare direcționată după contur - *Edge-directional interpolation*
- etc.

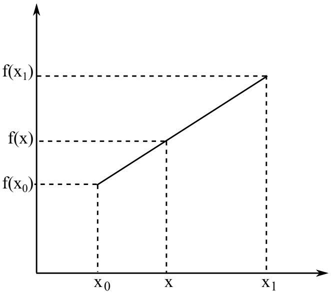
Interpolarea liniară



Interpolarea liniară prin două puncte $(x_0, f(x_0))$ și $(x_1, f(x_1))$:

$$\frac{f(x) - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0} \Leftrightarrow$$

Interpolarea liniară

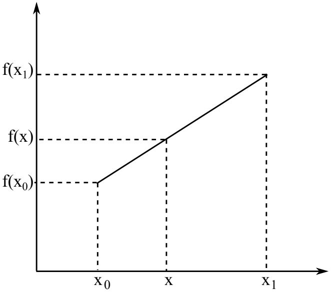


Interpolarea liniară prin două puncte $(x_0, f(x_0))$ și $(x_1, f(x_1))$:

$$\frac{f(x) - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0} \Leftrightarrow$$

$$f(x) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)(x - x_0) + f(x_0) =$$

Interpolarea liniară



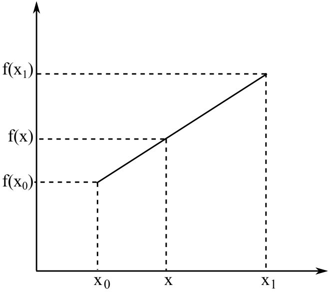
Interpolarea liniară prin două puncte $(x_0, f(x_0))$ și $(x_1, f(x_1))$:

$$\frac{f(x) - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0} \Leftrightarrow$$

$$f(x) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)(x - x_0) + f(x_0) =$$

$$= f(x_1)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0)\frac{(x_1 - x)}{(x_1 - x_0)} =$$

Interpolarea liniară



Interpolarea liniară prin două puncte $(x_0, f(x_0))$ și $(x_1, f(x_1))$:

$$\frac{f(x) - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0} \Leftrightarrow$$

$$f(x) = \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)(x - x_0) + f(x_0) =$$

$$= f(x_1)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0)\frac{(x_1 - x)}{(x_1 - x_0)} =$$

$$= f(x_1)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0)\left(1 - \frac{(x - x_0)}{(x_1 - x_0)}\right)$$

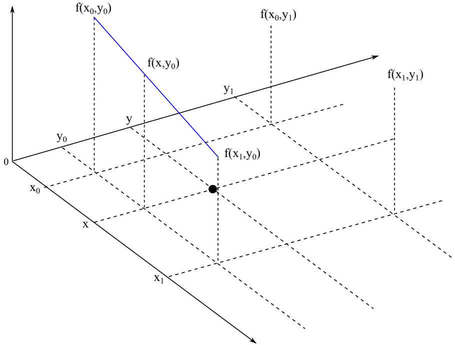
Interpolarea biliniară - Modelare matematică

Date de intrare:

- Patru puncte (x_0, y_0), (x_0, y_1), (x_1, y_0), (x_1, y_1), unde $x_0 < x_1$ și $y_0 < y_1$.
- Se cunosc valorile funcției în cele 4 puncte:

$$\begin{array}{ll} f(x_0, y_0) = f_{00} & f(x_0, y_1) = f_{01} \\ f(x_1, y_0) = f_{10} & f(x_1, y_1) = f_{11} \end{array}$$

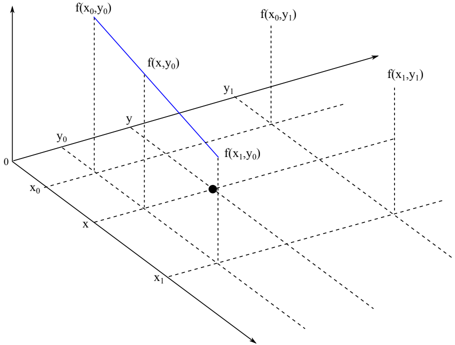
Interpolarea biliniară - Modelare matematică



Interpolare liniară

1. după axa $0x$ între (x_0, y_0) și (x_1, y_0):

Interpolarea biliniară - Modelare matematică

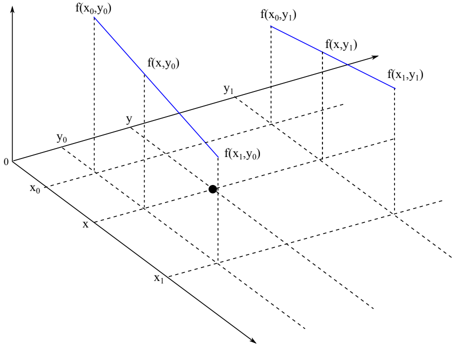


Interpolare liniară

1. după axa $0x$ între (x_0, y_0) și (x_1, y_0):

$$f(x, y_0) = f(x_1, y_0)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0, y_0)\left(1 - \frac{(x - x_0)}{(x_1 - x_0)}\right)$$

Interpolarea biliniară - Modelare matematică



Interpolare liniară

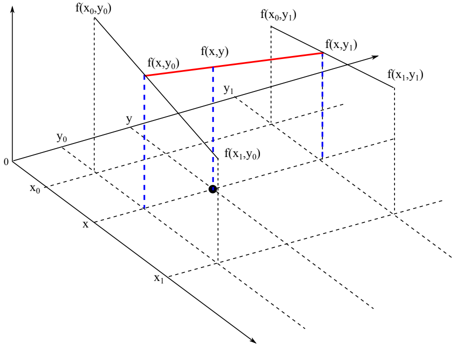
1. după axa $0x$ între (x_0, y_0) și (x_1, y_0):

$$f(x, y_0) = f(x_1, y_0)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0, y_0)\left(1 - \frac{(x - x_0)}{(x_1 - x_0)}\right)$$

2. după axa $0x$ între (x_0, y_1) și (x_1, y_1):

$$f(x, y_1) = f(x_1, y_1)\frac{(x - x_0)}{(x_1 - x_0)} + f(x_0, y_1)\left(1 - \frac{(x - x_0)}{(x_1 - x_0)}\right)$$

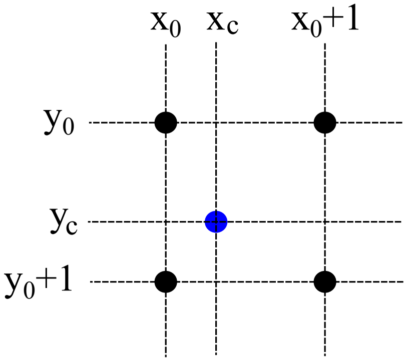
Interpolarea biliniară - Modelare matematică



3. Interpolare liniară după axa $0y$ între (x, y_0) și (x, y_1):

$$f(x,y) = f(x,y_1)\frac{(y-y_0)}{(y_1-y_0)} + f(x,y_0)\left(1-\frac{(y-y_0)}{(y_1-y_0)}\right)$$

Scalarea imaginilor - Interpolare



1. $\forall(x, y)$ din imaginea rezultat calculăm
$$x_c = \frac{x}{a}, y_c = \frac{y}{a}$$
2. definim $x_0 = [x_c], y_0 = [y_c]$;
3. definim $x_1 = x_0 + 1, y_1 = y_0 + 1$

Scalarea imaginilor - Interpolare

4. Interpolăm după $0x$.
   - Formula: $f(x, y_0) = f(x_1, y_0)\frac{(x-x_0)}{(x_1-x_0)} + f(x_0, y_0)\left(1 - \frac{(x-x_0)}{(x_1-x_0)}\right)$ devine

Scalarea imaginilor - Interpolare

4. Interpolăm după $0x$.
   - Formula: $f(x, y_0) = f(x_1, y_0)\frac{(x-x_0)}{(x_1-x_0)} + f(x_0, y_0)\left(1 - \frac{(x-x_0)}{(x_1-x_0)}\right)$ devine

$$f(x_c, [y_c]) = (\{x_c\})f([x_c] + 1, [y_c]) + (1 - \{x_c\})f([x_c], [y_c])$$

Scalarea imaginilor - Interpolare

4. Interpolăm după $0x$.
   - Formula: $f(x,y_0) = f(x_1,y_0)\frac{(x-x_0)}{(x_1-x_0)} + f(x_0,y_0)\left(1-\frac{(x-x_0)}{(x_1-x_0)}\right)$ devine

$$f(x_c,[y_c]) = (\{x_c\})f([x_c]+1,[y_c]) + (1-\{x_c\})f([x_c],[y_c])$$

   - Formula $f(x,y_1) = f(x_1,y_1)\frac{(x-x_1)}{(x_1-x_0)} + f(x_0,y_1)\left(1-\frac{(x-x_0)}{(x_1-x_0)}\right)$

Scalarea imaginilor - Interpolare

4. Interpolăm după $0x$.
   - Formula: $f(x, y_0) = f(x_1, y_0)\frac{(x-x_0)}{(x_1-x_0)} + f(x_0, y_0)\left(1 - \frac{(x-x_0)}{(x_1-x_0)}\right)$ devine
     $$f(x_c, [y_c]) = (\{x_c\})f([x_c]+1, [y_c]) + (1-\{x_c\})f([x_c], [y_c])$$
   - Formula $f(x, y_1) = f(x_1, y_1)\frac{(x-x_1)}{(x_1-x_0)} + f(x_0, y_1)\left(1 - \frac{(x-x_0)}{(x_1-x_0)}\right)$
     $$f(x_c, [y_c]+1) = (\{x_c\})f([x_c]+1, [y_c]+1) + (1-\{x_c\})f([x_c], [y_c]+1)$$

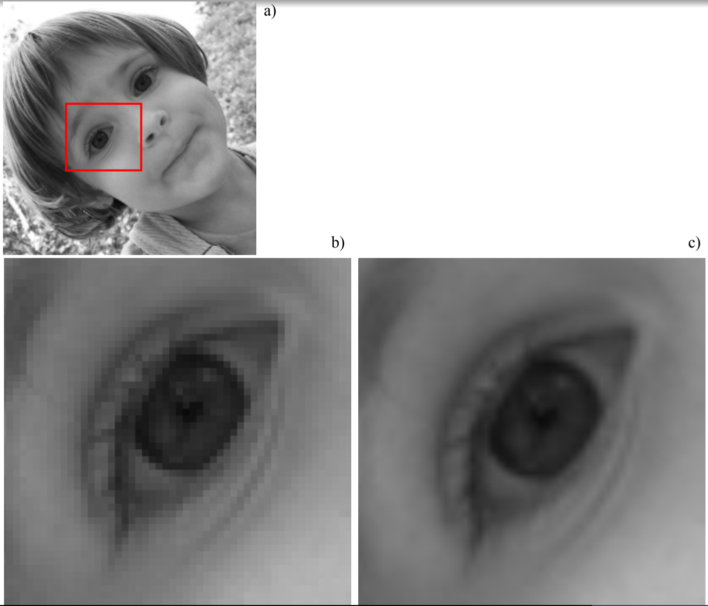
Scalarea imaginilor - Interpolare

5. Interpolăm după $0y$. Formula: $f(x,y) \approx f(x,y_1)\frac{(y-y_0)}{(y_1-y_0)} + f(x,y_0)\left(1 - \frac{(y-y_0)}{(y_1-y_0)}\right)$ devine:

$$f(x_c, y_c) = (\{y_c\})f(x_c, [y_c]+1) + (1-\{y_c\})f(x_c, [y_c])$$

Deci pixelul (x, y) din imaginea rezultat va avea culoarea calculată conform formulei de mai sus.

Scalarea imaginilor - Interpolare



(a) Imaginea sursă.
(b) Fragment din imaginea scalată cu factorul 5.
(c) Fragment din imaginea scalată cu factorul 5 cu interpolare biliniară.

Rotația imaginilor - Interpolare

Rotație cu unghiul θ în jurul punctului (x_{centru}, y_{centru})

Imagine rezultat $\longrightarrow$ imagine sursă: Pentru fiecare pixel (x', y') din imaginea rezultat, coordonate reale din imaginea sursă sunt date de:

$$\begin{cases} x_c = (x' - x_{centru}) \cos\theta + (y' - y_{centru}) \sin\theta + x_{centru} \\ y_c = -(x' - x_{centru}) \sin\theta + (y' - y_{centru}) \cos\theta + y_{centru} \end{cases}$$

Apoi se procedează cu interpolarea biliniară descrisă anterior.

Rotația imaginilor - Interpolare

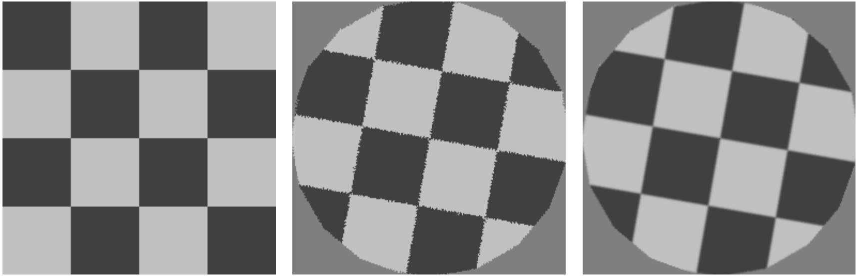

a) b) c)

Figura: a) Imagine de test; b) Imaginea rotită succesiv de 5 ori cu un unghi $\theta = 20^0$, cu rotunjirea valorilor coordonatelor calculate în imaginea sursă; c) Imaginea rotită succesiv de 5 ori cu un unghi $\theta = 20^0$, folosind interpolare biliniară;.

Interpolare prin convoluție

Problematică 1D:

- Cunoaștem $f(0), f(1), \ldots f(n)$
- Vrem $f(x)$ pentru orice $x \in [[x], [x]+1]$, x valoare reală și $[x]$ = partea întreagă a lui x

Interpolarea liniară:

$$f(x) = (f([x]+1) - f([x]))(x-[x]) + f([x]) = f([x]+1)(x-[x]) + f([x])(1+[x]-x)$$

Notăm cu $\{x\}$ partea fractionară a lui x , $\{x\} = x - [x] \Rightarrow$

$$f(x) = (1-\{x\})f([x]) + \{x\}f([x]+1)$$

Fiecare vecin contribuie cu o pondere invers proporțională cu distanța sa față de x

Interpolare prin convoluție

Problematică 1D: Interpolarea liniară între valori întregi consecutive
Formula:

$$f(x) = (1 - \{x\})f([x]) + \{x\}f([x] + 1)$$

poate fi s crisă sub formă de convoluție cu un *kernel* w

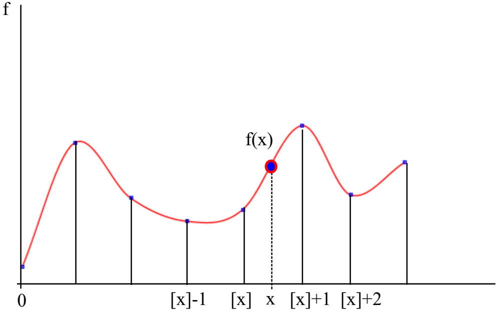
$$\overline{f}(x) = (w * f)(x) = \sum_{k=-\infty}^{\infty} w(x-k)f(k)$$

unde

$$w(x) = \begin{cases} 1 - x & \text{dacă } |x| < 1 \\ 0 & \text{dacă} |x| \geq 1 \end{cases}$$

Interpolare prin convoluție

Problematică 1D: Interpolarea cubică - se realizează considerând 4 vecini ai lui x.
În continuare consider cunoscută funcția în toate punctele discrete $f(0)$, $f(1) \ldots f(n)$.



În punctul $x \in [[x], [x]+1]$ $f(x)$ se calculează în funcție de f în cele 4 puncte vecine $[x]-1$, $[x]$, $[x]+1$ și $[x]+2$

$$\overline{f}(x) = (w_{cub} * f)(x) = \sum_{k=[x]-1}^{[x]+2} w_{cub}(x-k) f(k)$$

Interpolare prin convoluție

Problematică 1D: Interpolarea printr-un *spline* cubic - se realizează considerând 4 vecini ai lui x.

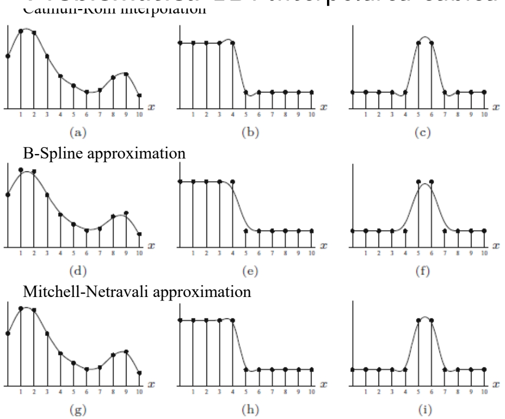
$$\overline{f}(x) = (w_{cs} * f)(x) = \sum_{k=[x]-1}^{[x]+2} w_{cs}(x-k)f(k)$$

Formula generală pentru w_{cs} cu parametrii a și b:

$$w_{cs}(t,a,b) = \begin{cases} (-a-1.5b+2)|t|^3 + & \\ +(a+2b-3)|t|^2 - (1/3)b + 1 & \text{dacă } 0 \leq |t| < 1 \\ (-a-b/6)|t|^3 + (5a+b)|t|^2 & \\ +(-8a-2b)|t| + 4a + (4/3)b & \text{dacă } 1 \leq |t| < 2 \\ 0 & \text{dacă } |t| \geq 2 \end{cases}$$

Interpolare prin convoluție

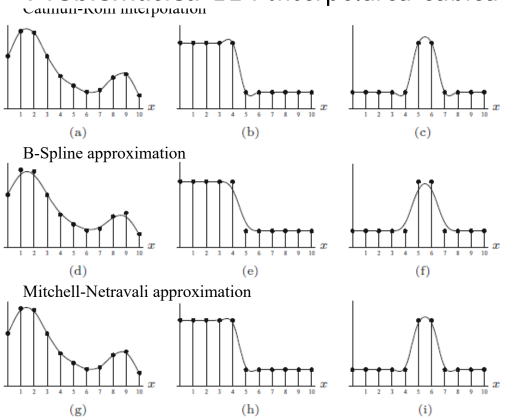
Problematică 1D: Interpolarea cubică



- **Curbe Catmull-Rom**: $a = 0.5$, $b = 0$

Interpolare prin convoluție

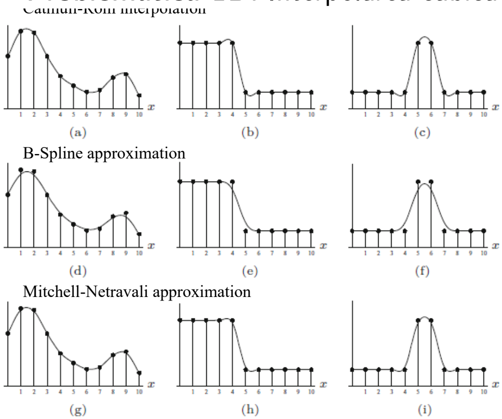
Problematică 1D: Interpolarea cubică



- **Curbe Catmull-Rom**: $a = 0.5$, $b = 0$
- **Cubic B-Splines**: $a = 0$, $b = 1$ - nu trec prin punctele de control $\Rightarrow$ este o aproximare!

Interpolare prin convoluție

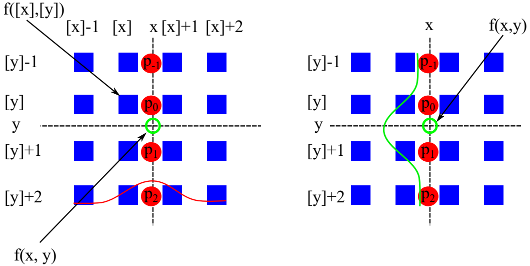
Problematică 1D: Interpolarea cubică



- **Curbe Catmull-Rom**: $a = 0.5$, $b = 0$
- **Cubic B-Splines**: $a = 0$, $b = 1$ - nu trec prin punctele de control $\Rightarrow$ este o aproximare!
- **Mitchell-Netravali approximation**: $a = 1/3$, $b = 1/3$ - nu trece prin punctele de control - oferă rezultate bune pentru imagini

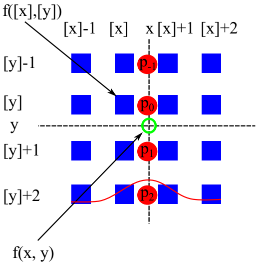
Interpolare prin convoluție

Problematică 2D: Interpolarea cubică



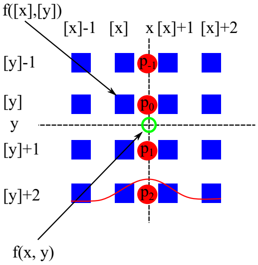
- se realizează întâi 4 interpolări pe orizontală $\Rightarrow$ se obțin 4 valori aproximate
- se realizeasă interpolarea între cele 4 valori obținute după direcția verticaă.

Interpolare prin convoluție



Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

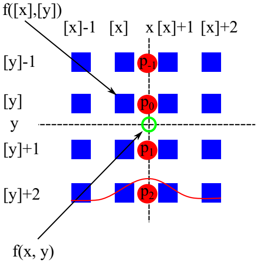
Interpolare prin convoluție



Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

$$p_i = w_{cs}(\{x\}+1)f([x]-1,[y]+i) + w_{cs}(\{x\})f([x],[y]+i) +$$
$$+w_{cs}(1-\{x\})f([x]+1,[y]+i) + w_{cs}(2-\{x\})f([x]+2,[y]+i) =$$

Interpolare prin convoluție



Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

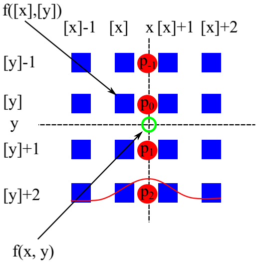
$$p_i = w_{cs}(\{x\}+1)f([x]-1,[y]+i) + w_{cs}(\{x\})f([x],[y]+i) +$$

$$+ w_{cs}(1-\{x\})f([x]+1,[y]+i) + w_{cs}(2-\{x\})f([x]+2,[y]+i) =$$

$$= \frac{1}{2}[(-\{x\}^3+2\{x\}^2-\{x\}+0)f([x]-1,[y]+i)+(3\{x\}^3-5\{x\}^2+0\{x\}+2)f([x],[y]+i)+$$

$$+(-3\{x\}^3+4\{x\}^2+\{x\}+0)f([x]+1,[y]+i)+(\{x\}^3-\{x\}^2+0\{x\}+0)f([x]+,[y]+i)]$$

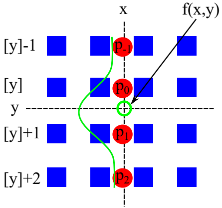
Interpolare prin convoluție



Practic formulele pot fi sintetizate sub formă de produs matricial

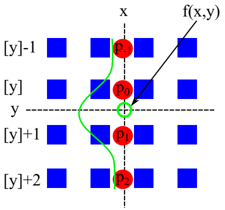
$$p_i = \frac{1}{2}\begin{bmatrix} 1 & \{x\} & \{x\}^2 & \{x\}^3 \end{bmatrix} \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} f([x]-1,[y]+i) \\ f([x],[y]+i) \\ f([x]+1,[y]+i) \\ f([x]+2,[y]+i) \end{bmatrix}$$

Interpolare prin convoluție



Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

Interpolare prin convoluție

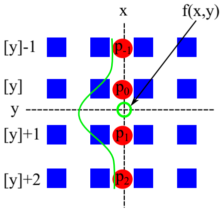


Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

$$f(x,y) = w_{cs}(\{y\}+1)p_{-1} + w_{cs}(\{y\})p_0 +$$

$$+ w_{cs}(1-\{y\})p_1 + w_{cs}(2-\{y\})p_2 =$$

Interpolare prin convoluție

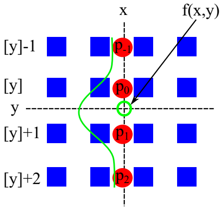


Practic folosind notațiile din figură și considerând varianta Catmull-Rom cu $a = 0,5$, $b = 0$, pentru $\forall i \in \{-1, 0, 1, 2\}$

$$f(x,y) = w_{cs}(\{y\}+1)p_{-1} + w_{cs}(\{y\})p_0 +$$
$$+ w_{cs}(1-\{y\})p_1 + w_{cs}(2-\{y\})p_2 =$$

$$= \frac{1}{2}[(-\{y\}^3 + 2\{y\}^2 - \{y\} + 0)p_{-1} + (3\{y\}^3 - 5\{y\}^2 + 0\{y\} + 2)p_0 +$$
$$+ (-3\{y\}^3 + 4\{y\}^2 + \{y\} + 0)p_1 + (\{y\}^3 - \{y\}^2 + 0\{y\} + 0)p_2]$$

Interpolare prin convoluție



Practic formula paote fi sintetizată sub formă de produs matricial

$$f(x,y) = \frac{1}{2}\begin{bmatrix} 1 & \{y\} & \{y\}^2 & \{y\}^3 \end{bmatrix}\begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix}\begin{bmatrix} p_{-1} \\ p_0 \\ p_1 \\ p_2 \end{bmatrix}$$

Tranformările geometrice ca produs de matrice

- **Rotația:**

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- **Scalarea:**

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

- **Translația:** nu poate fi reprezentată ca produs de matrice în sistemul $x0y$.

Coordonate omogene

Extinderea vectorilor bidimensionali cu o a treia componentă:

$$\vec{v} = \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \hat{x} \\ \hat{y} \\ h \end{pmatrix} = \begin{pmatrix} hx \\ hy \\ h \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ y_1 \\ h_1 \end{pmatrix} \equiv \begin{pmatrix} x_2 \\ y_2 \\ h_2 \end{pmatrix} \equiv \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Leftrightarrow \left(x = \frac{x_1}{h_1} = \frac{x_2}{h_2} \text{ și } y = \frac{y_1}{h_1} = \frac{y_2}{h_2} \right)$$

Transformările geometrice în coordonate omogene

- Translația cu dx, dy:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Rotația cu α:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Scalarea cu s_x, s_y:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformări afine

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

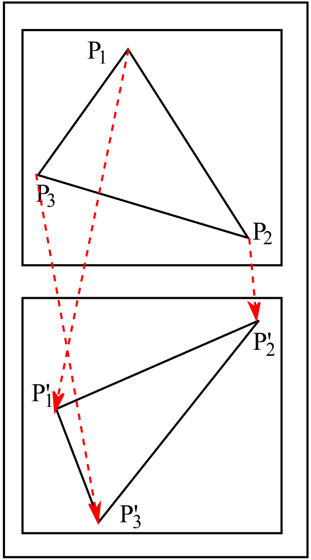
Scalarea & Rotația:

$$\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$

Translația:

$$\begin{pmatrix} a_{02} \\ a_{12} \end{pmatrix}$$

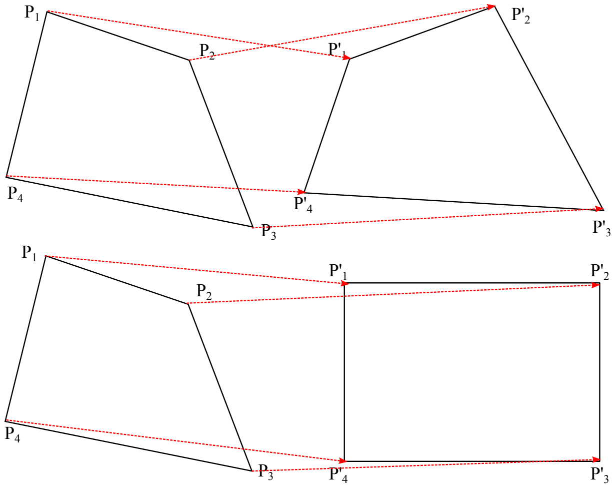
Transformări afine



Proprietăți:

- Transformă drepte în drepte.
- Transformă dreptunghiuri în paralelipipede.
- Păstrează rapoartele între distanțe egale.

Transformări proiective - Problematică



Observații:

- NU poate fi obținută printr-o transformare afină.
- Este necesară o transformare *proiectivă*:

$$\begin{pmatrix} hx' \\ hy' \\ h \end{pmatrix} = A \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

unde

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

Transformări proiective - Fundamentare matematică

Din

$$\begin{pmatrix} hx' \\ hy' \\ h \end{pmatrix} = A \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

rezultă:

$$h_1 P_1' = A * P_1 \quad h_3 P_3' = A * P_3$$

$$h_2 P_2' = A * P_2 \quad h_4 P_4' = A * P_4$$

unde

$$P_i = \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}, P_i' = \begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix}, i = \overline{1, 4}$$

Transformări proiective - Fundamentare matematică

În plus P_1, P_2, P_3, P_4 - liniar dependete și P'_1, P'_2, P'_3, P'_4 - liniar dependente $\Rightarrow$

$$P_4 = b_1 P_1 + b_2 P_2 + b_3 P_3 = (P_1\ P_2\ P_3) \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = P * b$$

$$P'_4 = b'_1 P'_1 + b'_2 P'_2 + b'_3 P'_3 = (P'_1\ P'_2\ P'_3) \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \end{pmatrix} = \begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b'_1 \\ b'_2 \\ b'_3 \end{pmatrix} = P' * b'$$

Transformări proiective - Fundamentare matematică

Avem:

- $P_4 = P * b$ și $P'_4 = P' * b'$ cu $P = (P_1\ P_2\ P_3), P' = (P'_1\ P'_2\ P'_3)$.
- P_1, P_2, P_3 - liniar independente și P'_1, P'_2, P'_3 - liniar independente.

Rezultă: P, P' inversabile

Deci:

$$b = P^{-1} * P_4$$

$$b' = (P')^{-1} * P'_4$$

Transformări proiective - Fundamentare matematică

Calcul h_1, h_2, h_3 în funcție de h_4:

$$A * P_4 = A * P * b = A * (P_1 b_1 + P_2 b_2 + P_3 b_3) = \sum_{i=1}^{3} A * P_i * b_i = \sum_{i=1}^{3} h_i * P'_i * b_i$$

și

$$A * P_4 = h_4 * P'_4 = h_4 * (P'_1 b'_1 + P'_2 b' + P'_3 b'_3) = \sum_{i=1}^{3} h_4 * P'_i * b'_i$$

O soluție a sistemului:

$$\sum_{i=1}^{3} h_i * P'_i * b_i = \sum_{i=1}^{3} h_4 * P'_i * b'_i$$

este:

$$h_i = \frac{b'_i}{b_i} h_4, i = 1, 2, 3$$

Transformări proiective - Fundamentare matematică

Matricea A de transformare:

$$A * P = A * (P_1 \ P_2 \ P_3) = (h_1 * P'_1, \ \ h_2 * P'_2, \ \ h_3 * P'_3) = h_4 * \begin{pmatrix} \frac{b'_1}{b_1} P'_1 & \frac{b'_2}{b_2} P'_2 & \frac{b'_3}{b_3} P'_3 \end{pmatrix}$$

De unde:

$$A = h_4 * \begin{pmatrix} \frac{b'_1}{b_1} P'_1 & \frac{b'_2}{b_2} P'_2 & \frac{b'_3}{b_3} P'_3 \end{pmatrix} P^{-1}$$

Iar $h_4 \neq 0$ arbitrar. Se consideră $h_4 = 1$.

Transformări proiective - Algoritm

1. Se definesc punctele $P_1, P_2, P_3 P_4$ și P_1', P_2', P_3', P_4' a. î. cele două patrulatere să nu fie degenerate.

Transformări proiective - Algoritm

1. Se definesc punctele $P_1, P_2, P_3 P_4$ și P'_1, P'_2, P'_3, P'_4 a. î. cele două patrulatere să nu fie degenerate.
2. Se calculează vectorii b și b'.

Transformări proiective - Algoritm

1. Se definesc punctele $P_1, P_2, P_3 P_4$ și P_1', P_2', P_3', P_4' a. î. cele două patrulatere să nu fie degenerate.
2. Se calculează vectorii b și b'.
3. Se consideră $h_4 = 1$ și se calculează matricea de transformare A.

Transformări proiective - Algoritm

1. Se definesc punctele $P_1, P_2, P_3 P_4$ și P_1', P_2', P_3', P_4' a. î. cele două patrulatere să nu fie degenerate.
2. Se calculează vectorii b și b'.
3. Se consideră $h_4 = 1$ și se calculează matricea de transformare A.
4. Pentru fiecare pixel (x, y) reprezentat prin $(x, y, 1,)^T$ din imaginea originală se determină pixelul corespondent:

$$\begin{pmatrix} \hat{x}' \\ \hat{y}' \\ h \end{pmatrix} = A * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Pixelul corespunzător lui (x, y) din imaginea rezultat va fi $\left(\hat{x}'/h, \hat{y}'/h\right)$

Transformări proiective - Exemplu

Transformări proiective - Interpolare

- **Problemă:** - prin transformarea de la imaginea sursă către imaginea rezultat apar pixeli în care lipsește informația!
- **Soluție:** - transformarea rezultat $\rightarrow$ sursă $+$ interpolare:
  - Se consideră punctele P_1, P_2, P_3, P_4 în imaginea rezultat și P_1', P_2', P_3', P_4' în imaginea sursă

Transformări proiective - Interpolare

- **Problemă:** - prin transformarea de la imaginea sursă către imaginea rezultat apar pixeli în care lipsește informația!
- **Soluție:** - transformarea rezultat $\rightarrow$ sursă $+$ interpolare:
  - Se consideră punctele P_1, P_2, P_3, P_4 în imaginea rezultat și P'_1, P'_2, P'_3, P'_4 în imaginea sursă
  - Cu aceste considerente se calculează matricea A .

Transformări proiective - Interpolare

- **Problemă:** - prin transformarea de la imaginea sursă către imaginea rezultat apar pixeli în care lipsește informația!

- **Soluție:** - transformarea rezultat $\rightarrow$ sursă $+$ interpolare:
  - Se consideră punctele P_1, P_2, P_3, P_4 în imaginea rezultat și P_1', P_2', P_3', P_4' în imaginea sursă
  - Cu aceste considerente se calculează matricea A .
  - Pentru fiecare pixel $(x', y', 1)$ din imaginea rezultat se calculează pixelul de proveniență:$(x_s, y_s, h_s)^T = A * (x', y', 1)^T$ din imaginea sursă $(x_s, y_s, h_s =$ valori reale).

Transformări proiective - Interpolare

- **Problemă:** - prin transformarea de la imaginea sursă către imaginea rezultat apar pixeli în care lipsește informația!

- **Soluție:** - transformarea rezultat $\rightarrow$ sursă + interpolare:
  - Se consideră punctele P_1, P_2, P_3, P_4 în imaginea rezultat și P_1', P_2', P_3', P_4' în imaginea sursă
  - Cu aceste considerente se calculează matricea A .
  - Pentru fiecare pixel $(x', y', 1)$ din imaginea rezultat se calculează pixelul de proveniență:$(x_s, y_s, h_s)^T = A * (x', y', 1)^T$ din imaginea sursă $(x_s, y_s, h_s =$ valori reale).
  - Se normează: $x_c = x_s/h_s$, $y_c = y_s/h_s$.

Transformări proiective - Interpolare

- **Problemă:** - prin transformarea de la imaginea sursă către imaginea rezultat apar pixeli în care lipsește informația!
- **Soluție:** - transformarea rezultat $\rightarrow$ sursă $+$ interpolare:
  - Se consideră punctele P_1, P_2, P_3, P_4 în imaginea rezultat și P'_1, P'_2, P'_3, P'_4 în imaginea sursă
  - Cu aceste considerente se calculează matricea A .
  - Pentru fiecare pixel $(x', y', 1)$ din imaginea rezultat se calculează pixelul de proveniență:$(x_s, y_s, h_s)^T = A * (x', y', 1)^T$ din imaginea sursă ($x_s, y_s, h_s =$ valori reale).
  - Se normează: $x_c = x_s/h_s$, $y_c = y_s/h_s$.
  - Se folosesc formulele de interpolare pentru a calcula valoearea pixelului (x', y') din valorile pixelilor $([x_c], [y_c])$, $([x_c] + 1, [y_c])$, $([x_c], [y_c] + 1)$, $([x_c] + 1, [y_c] + 1)$.

Transformări proiective - Exemplu

Transformări proiective - Exemplu

Rezumat

- Transformările geometrice de bază: translația, rotația, scalarea.
- Interpolarea biliniară.
  - Problematica interpolării.
  - Interpolarea liniară - interpolarea biliniară.
  - Interpolarea cubică
  - Aplicarea interpolării pentru imagini după scalare și rotație.
- Transformarea proiectivă.
  - Reprezentarea în coordonate omogene.
  - Transformări afine.
  - Transformări proiective - colonierea proiectivă.