



Deformace rastrových obrázků

© 1997-2011 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
<http://cgg.mff.cuni.cz/~pepca/>

Deformace obrázků - “warping”



- ✦ **mapování textur** při 3D zobrazování
 - perspektivní zkreslení, mapování na oblá tělesa
- ✦ **oprava geometrického zkreslení** při digitalizaci obrazu
 - družicové a letecké snímky, scanování nerovného dokumentu, ..
- ✦ **speciální efekty** v televizním a filmovém průmyslu, reklamě a DTP

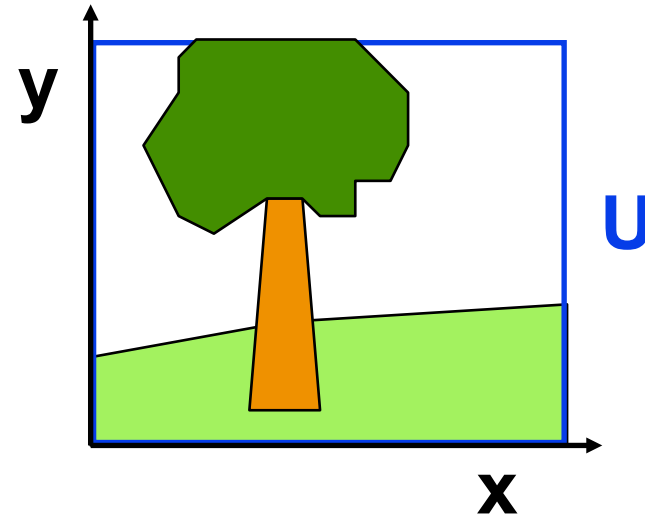


Matematická terminologie

„obrazová funkce”

$$f: U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$$

$$f: [x, y] \rightarrow [a_1, a_2, \dots, a_n]$$



poloha bodu
v rovině

atributy obrazu
(barva, průhlednost)



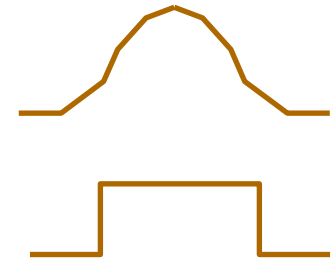
Diskrétní reprezentace

Digitální rastrový obraz:

$$I: \langle 0..m-1 \rangle \times \langle 0..n-1 \rangle \rightarrow \mathbb{R}^n$$

Digitalizace pomocí filtru **d**:

$$I_f(i, j) = \iint_{\mathbb{R}^2} f(x, y) \cdot \underline{d(x-i, y-j)} \, dx \, dy$$



d vyjadřuje snímací charakteristiku digitalizačního zařízení (fotočidlo, CCD prvek)



Zobrazení digitálního obrazu

Rekonstrukce diskrétního obrazu:

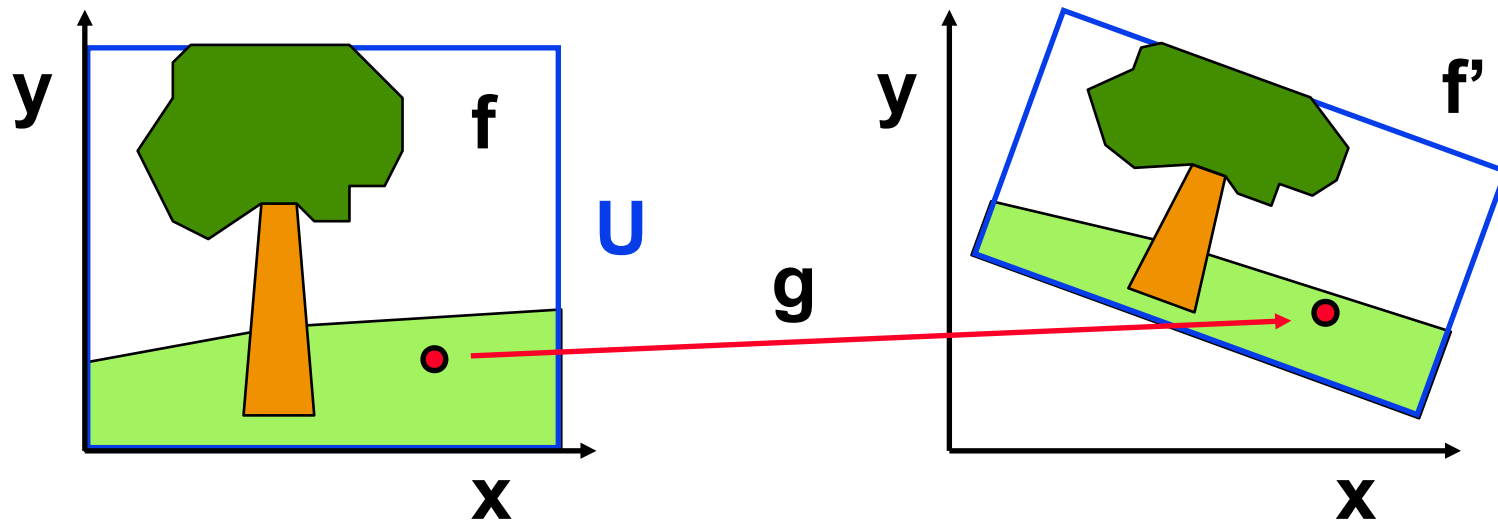
$$\mathbf{f}^r(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{l}_f(i, j) \cdot \mathbf{r}(i - \mathbf{x}, j - \mathbf{y})$$

\mathbf{r} charakterizuje výstupní zařízení (pulsní odezva, stopa jednoho pixelu, ..)

Chceme, aby se \mathbf{f}^r co nejvíce podobala \mathbf{f}
- v oblasti frekvencí, které se dají zachytit
při zvolené hustotě vzorkování (Nyquist)



Geometrická transformace



$$g: \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

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

$$f(g^{-1}(u, v)) = f'(u, v)$$

Co s rastrovými obrázky?

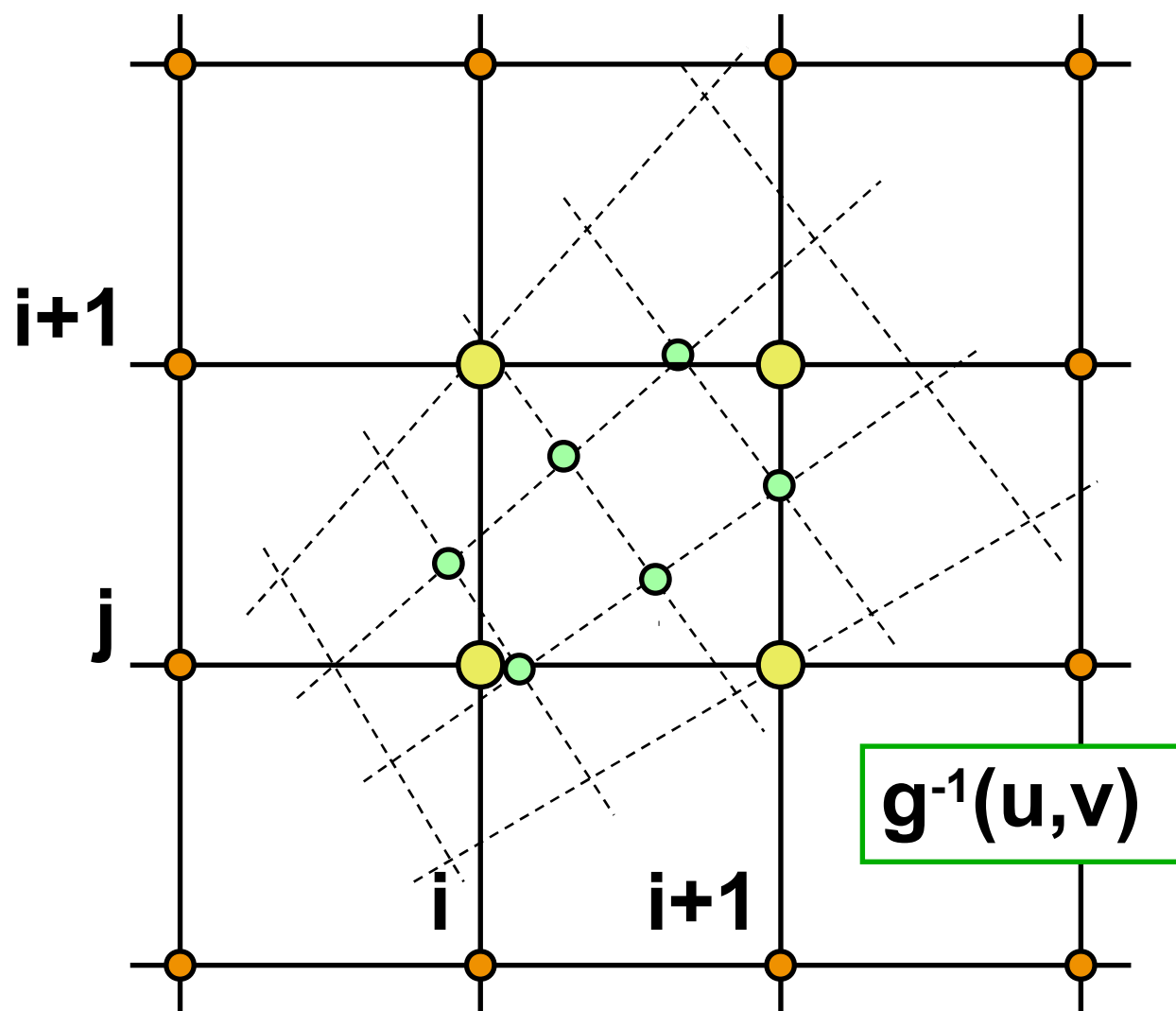
$$G(I_f) \cong I_{f'}$$



Transformace s interpolací

- digitalizace = **vzorkování**
 - digitalizační filtr má pouze bodový nosič (Diracův impuls)
- výpočet atributů (barvy) transformovaného pixelu pomocí **interpolace** nebo **aproximace**
 - je potřeba inverzní transformační funkce \mathbf{g}^{-1}
- „zaokrouhlení“, **polynomiální interpolace**
 - v praxi se nejvíce používá bilineární až bikubická interpolace/aproximace

Interpolace ve zdrojových souřad.



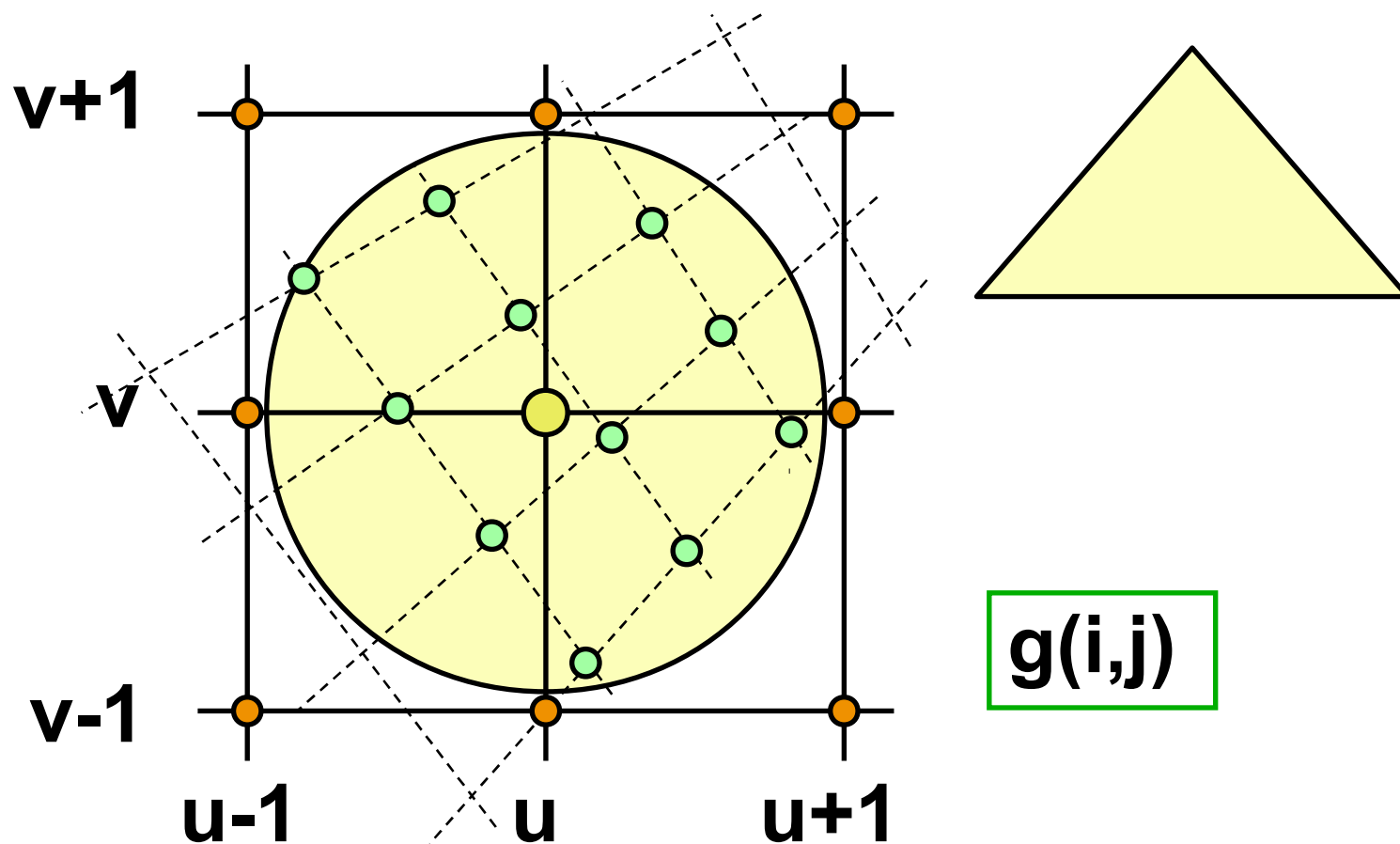


Transformace s filtrací

- odpovídá představě **plošného pixelu**
 - digitalizační filtr má nosič s nenulovou plochou (např. obdélníkový nebo konický filtr)
- zdrojové pixely se přenášejí do prostoru **výsledného obrázku**
 - je potřeba pouze transformační funkce **g**
- vhodné i pro **kontraktivní transformace**
 - při izometrickém zobrazení se obrázek rozmazává
 - velká kontrakce \Rightarrow velká výpočetní náročnost



Transformace s filtrací

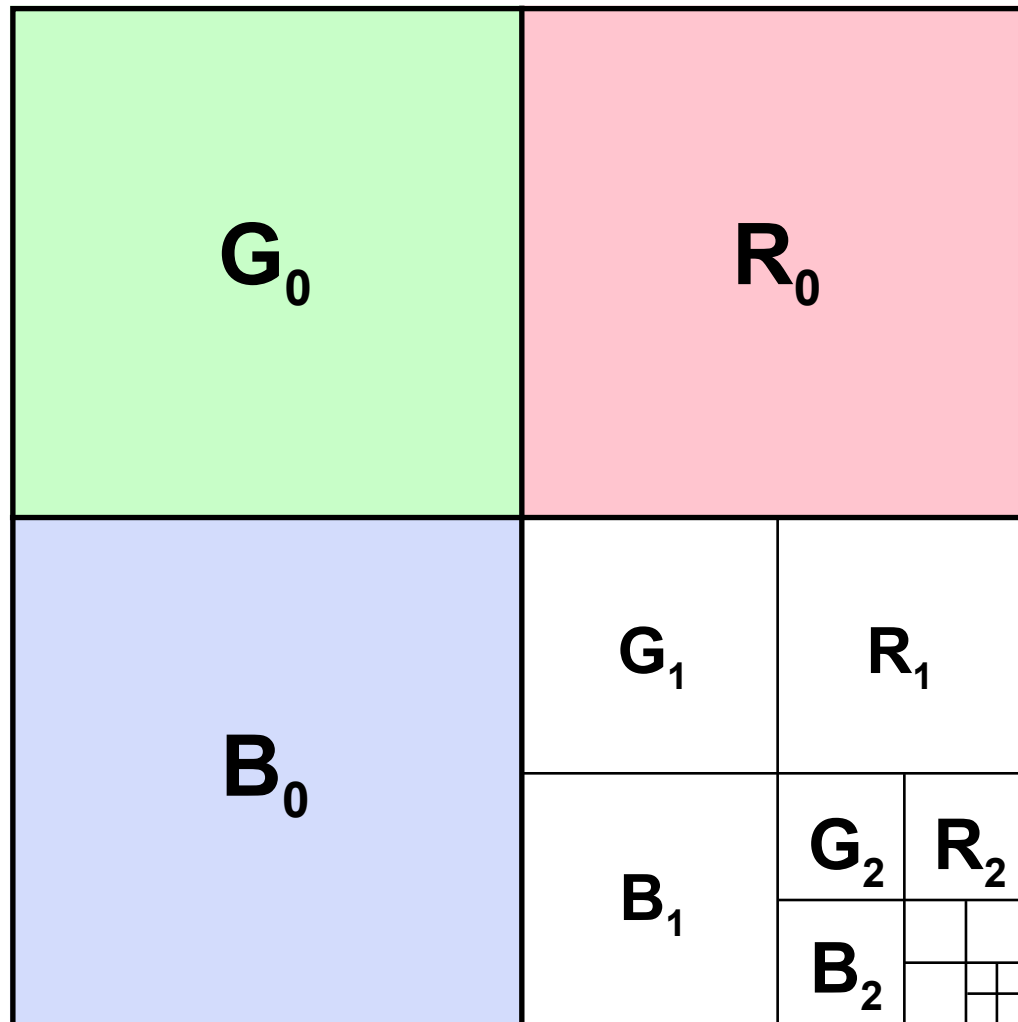




“MIP map” (*multum in parvo*)

- **pyramidální reprezentace** zdrojového obrázku (hierarchická reprezentace)
 - předzpracování = předfiltrování
- výhodná zejména při **silnější kontrakci**
 - úspora času výpočtu při filtracích
 - používá se zejména při mapování textur (mapování na vzdálenější objekty)
- **kompaktní reprezentace** v paměti
 - zabírá pouze **4/3** paměti RGB obrázku

“MIP map”



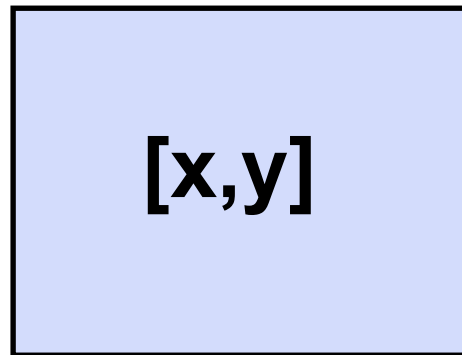


Vícekové transformace

- rozložení transformace do **několika kroků**
 - každý krok operuje pouze na řádcích nebo pouze na sloupcích
- **rychlejší výpočet**
 - jednodušší filtrace v řádcích/sloupcích
 - dvě 1D filtrace jsou rychlejší než jedna 2D
- **„bottleneck problem“** - ztráta přesnosti mezivýsledku
 - je-li částečné zobrazení hodně kontraktivní nebo dokonce neinjektivní



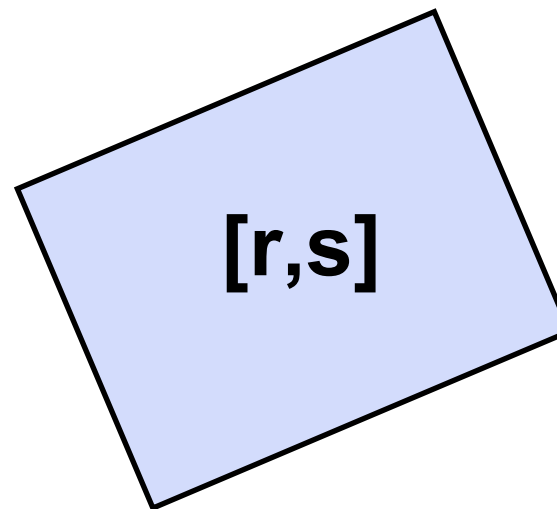
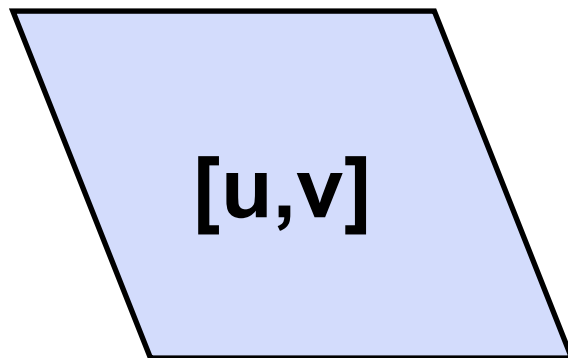
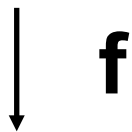
Otáčení ve dvou krocích



$$[u,v] = f(x,y) = [f_1(x,y), y]$$

$$[r,s] = g(u,v) = [u, g_1(u,v)]$$

$$= [f_1(x,y), g_1(f_1(x,y), v)]$$





Odvození

Cílová transformace (otočení o úhel α):

$$\begin{aligned} \mathbf{r} &= \mathbf{x} \cdot \cos \alpha - \mathbf{y} \cdot \sin \alpha \\ \mathbf{s} &= \mathbf{x} \cdot \sin \alpha + \mathbf{y} \cdot \cos \alpha \end{aligned}$$

První transformace \mathbf{f} :

$$\underline{\mathbf{u} = \mathbf{f}_1(\mathbf{x}, \mathbf{y})} \quad \underline{\mathbf{v} = \mathbf{y}}$$

Druhá transformace \mathbf{g} :

$$\underline{\mathbf{r} = \mathbf{u} = \mathbf{f}_1(\mathbf{x}, \mathbf{y})} \quad \underline{\mathbf{s} = \mathbf{g}_1(\mathbf{u}, \mathbf{v})} = \mathbf{g}_1[\mathbf{f}_1(\mathbf{x}, \mathbf{y}), \mathbf{y}]$$



Odvození

Horizontální zkosení v prvním kroku:

$$\mathbf{f}_1(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \cos\alpha - \mathbf{y} \cdot \sin\alpha$$

Vertikální zkosení v druhém kroku:

$$\mathbf{g}_1(\mathbf{u}, \mathbf{v}) = \mathbf{x} \cdot \sin\alpha + \mathbf{y} \cdot \cos\alpha = \mathbf{x} \cdot \sin\alpha + \mathbf{v} \cdot \cos\alpha$$

$$\mathbf{u} = \mathbf{f}_1(\mathbf{x}, \mathbf{v}) = \mathbf{x} \cdot \cos\alpha - \mathbf{v} \cdot \sin\alpha$$

$$\mathbf{x} = (\mathbf{u} + \mathbf{v} \cdot \sin\alpha) / \cos\alpha$$

$$\mathbf{g}_1(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \tan\alpha + \mathbf{v} \cdot \sec\alpha$$



Tříkrokové otočení

Rozklad **matic** **otočení** na tři činitele:

$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & -\tan \alpha / 2 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ \sin \alpha & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -\tan \alpha / 2 \\ 0 & 1 \end{bmatrix}$$

vhodné pro úhly otočení

$$-\pi/2 \leq \alpha \leq \pi/2$$



Otočení o obecný úhel

- **vícekrokové algoritmy** nedávají kvalitní výsledky pro všechny úhly. Vhodné rozsahy α :
 - $45^\circ \leq \alpha \leq 45^\circ$ pro dvoukrokovou metodu
 - $90^\circ \leq \alpha \leq 90^\circ$ pro tříkrokovou metodu
- **pomocné otočení o násobek 90°**
 - jednoduchý výpočet bez nutnosti filtrování a bez ztráty přesnosti

Obecná separabilní transformace

Separabilní transformace:

$$\mathbf{h}(\mathbf{x}, \mathbf{y}) = [\mathbf{h}_1(\mathbf{x}, \mathbf{y}), \mathbf{h}_2(\mathbf{x}, \mathbf{y})] = \underline{\mathbf{g}(\mathbf{f}(\mathbf{x}, \mathbf{y}))}$$

$$\underline{\mathbf{f}(\mathbf{x}, \mathbf{y})} = [\mathbf{f}_1(\mathbf{x}, \mathbf{y}), \mathbf{y}]$$

$$\underline{\mathbf{g}(\mathbf{u}, \mathbf{v})} = [\mathbf{u}, \mathbf{g}_1(\mathbf{u}, \mathbf{v})]$$

$$\mathbf{f}_1(\mathbf{x}, \mathbf{y}) = \mathbf{h}_1(\mathbf{x}, \mathbf{y})$$

$$\mathbf{g}_1(\mathbf{u}, \mathbf{v}) = \mathbf{h}_2(\varphi(\mathbf{u}, \mathbf{v}), \mathbf{v})$$

existuje-li $\varphi(\mathbf{u}, \mathbf{v})$, že

$$\underline{\mathbf{x} = \varphi(\mathbf{u}, \mathbf{v})}$$



Ztráty přesnosti

Nevýhodné jsou oblasti s malou hodnotou derivace

$$\frac{\partial \mathbf{h}_1(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}$$

nebo velkou hodnotou derivace

$$\frac{\partial \mathbf{g}_1(\mathbf{u}, \mathbf{v})}{\partial \mathbf{v}}$$

Někde ani nemusí být definována inverzní funkce $\varphi(\mathbf{u}, \mathbf{v})$!

\Rightarrow pomůže současné zpracování obrazů \mathbf{I} a \mathbf{I}^T



Optimalizovaný algoritmus

- ♦ **současné zpracování** původního (I) a transponovaného (I^T) obrazu
 - oba obrazy se nezávisle deformují ve dvou krocích (pomocí separabilních transformací h a h^T)
- ♦ výsledek se **pixel po pixelu** skládá z $h(I)$ a $(h^T(I^T))^T$
 - v každém pixelu se porovnávají míry degradace obou výsledných obrazů
- ♦ lze provádět i deformace, které **nejsou prosté**
 - „přehnutí“ obrázku



Další informace:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes:**
Computer Graphics, Principles and Practice, 815-832
- **J. Gomes et al.:** *Warping and Morphing of Graphical Objects*, Course Notes - SIGGRAPH'95