

7. Image Denoising

Készítette: Erdélyi András

A feladat

- Probléma: Egy zajos képnek a magas a Total Variation-je
- Cél: A Total Variaton csökkentése
 - $V(u) = \sum_{i,j} |u_{i+1,j} - u_{i,j}| + |u_{i,j+1} - u_{i,j}|$
- Feladat: Energiafüggvény minimalizálása
 - $\min_u \lambda V(u) + \frac{1}{2} \sum_{i,j} u_{i,j} - f_{i,j}$
 - Az első tag a TV, megtartja az éleket
 - A második tag biztosítja, hogy ne térjen el túlságosan az eredeti képtől

Gradiens módszer

- Az energiafüggvény minimalizálása iteratívan a gradiens csökkentésével

- k -adik iteráció, az i, j pixelre:

- $u^{(k+1)}(i, j) = u^{(k)}(i, j) - \alpha \left[\left(u^{(k)}(i, j) - f(i, j) \right) + \lambda \nabla \cdot \left(\frac{\nabla u^{(k)}(i, j)}{|\nabla u^{(k)}(i, j)|} \right) \right]$

- $\nabla u(i, j) \approx \begin{bmatrix} u(i+1, j) - u(i, j) \\ u(i, j+1) - u(i, j) \end{bmatrix}$

- $\nabla \cdot (v(i, j)) \approx (v_x(i, j) - v_x(i-1, j)) + (v_y(i, j) - v_y(i, j-1))$

Párhuzamosítás

- Minden számítás külön szálra kerül
- A különböző rész számítások szekvenciálisan
- Mindegyik iterációban a részsámítások kernel hívásai
- Iteráció végén szinkronizáció

Párhuzamosítás

- Gradiensek (∇u) kiszámítása pixelenként

- $\nabla u(i, j) \approx \begin{bmatrix} u(i+1, j) - u(i, j) \\ u(i, j+1) - u(i, j) \end{bmatrix}$

- Forward difference

- Divergencia ($\nabla \cdot (v)$) kiszámítása pixelenként

- $\nabla \cdot (v(i, j)) \approx (v_x(i, j) - v_x(i-1, j)) + (v_y(i, j) - v_y(i, j-1))$

- $v_x = \frac{u_x}{|\nabla u|}, v_y = \frac{u_y}{|\nabla u|}$

- Backward difference

- Píxelek frissítése

	$i, j - 1$			
$i - 1, j$	i, j	$i + 1, j$		
	$i, j + 1$			

Osztott memória

- Minden block-nak saját osztott memória
- A szálak gyorsabban eléri a szomszédos pixeleket
- Block-hoz képest minden oldalon plusz egy pixel halo