

Odstraňovanie tieňov

Spracovanie farebného obrazu

Rastislav Kamenický, Michal Piovarči, Matej Kopernický

github.com/misop/shadow_removal

- Široké využitie pri úprave fotografií



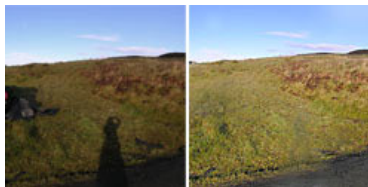
- ▶ Široké využitie pri úprave fotografií
- ▶ Neexistuje žiadne 100% riešenie



- ▶ Široké využitie pri úprave fotografií
- ▶ Neexistuje žiadne 100% riešenie
- ▶ Väčšinou sú vyžadované parametre snímacieho zariadenia, vlastnosti scény a pod.



- ▶ Široké využitie pri úprave fotografií
- ▶ Neexistuje žiadne 100% riešenie
- ▶ Väčšinou sú vyžadované parametre snímacieho zariadenia, vlastnosti scény a pod.
- ▶ Cieľ – vytvoriť jednoduchú metódu, ktorej vstupom je len obrázok a je potrebné len minimum interakcie



► Hlavný zdroj:

G. D. Finlayson, M. S. Drew, C. Lu: *Intrinsic Images by Entropy Minimization*, European Conference on Computer Vision, Vol. 3 (2004), pp. 582-595

- ▶ Hlavný zdroj:

G. D. Finlayson, M. S. Drew, C. Lu: *Intrinsic Images by Entropy Minimization*, European Conference on Computer Vision, Vol. 3 (2004), pp. 582-595

- ▶ Intrinsic image: obrázok nezávislý od svetla, a teda aj zbavený tieňov

- ▶ Hlavný zdroj:

G. D. Finlayson, M. S. Drew, C. Lu: *Intrinsic Images by Entropy Minimization*, European Conference on Computer Vision, Vol. 3 (2004), pp. 582-595

- ▶ Intrinsic image: obrázok nezávislý od svetla, a teda aj zbavený tieňov
- ▶ Entropia: „neusporiadanosť, zložitosť“ obrázku

- ▶ Najst' intrinsic obrázok

Postup

- ▶ Nájsť intrinsic obrázok
- ▶ Na základe neho identifikovať tieň

Postup

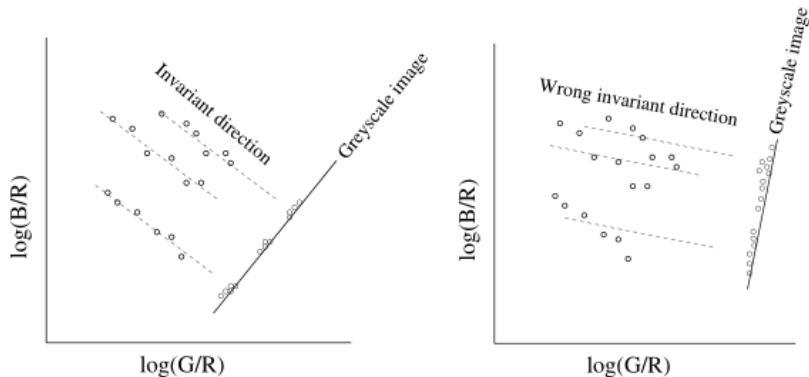
- ▶ Nájsť intrinsic obrázok
- ▶ Na základe neho identifikovať tieň
- ▶ Odstrániť tieň

Chromaticity

- ▶ 2D chromaticity colour space $\{\log(G/R), \log(B/R)\}$

Chromaticity

- ▶ 2D chromaticity colour space $\{\log(G/R), \log(B/R)\}$
!!!!!!!!!!!!!!!!!!!!!!chromaticity2
- ▶ Zmena v osvetlení v praxi zodpovedá posunu v chromatickom priestore jedným smerom (za predpokladu jedného zdroju svetla)



Hľadanie orientácie

- Projekciou bodov na priamku vo vhodnom smere θ získame šedoúrovňový obrázok

$$\mathcal{I} = \chi_1 \cos\theta + \chi_2 \sin\theta$$

Hľadanie orientácie

- Projekciou bodov na priamku vo vhodnom smere θ získame šedoúrovňový obrázok

$$\mathcal{I} = \chi_1 \cos\theta + \chi_2 \sin\theta$$

- Najvhodnejší smer θ nájdeme minimalizáciou entropie

$$-\sum_i p(x_i) * \log p(x_i)$$

Výpočet entropie

- ▶ Nepresnosti v reálnych fotkách – vstupný obrázok vyhladiťme Gaussianom

Výpočet entropie

- ▶ Nepresnosti v reálnych fotkách – vstupný obrázok vyhladíme Gaussianom
- ▶ Vytvoríme histogram šedoúrovňového obrázku, vezmeme iba 90% stredných dát

Výpočet entropie

- ▶ Nepresnosti v reálnych fotkách – vstupný obrázok vyhladíme Gaussianom
- ▶ Vytvoríme histogram šedoúrovňového obrázku, vezmeme iba 90% stredných dát
- ▶ „Bin width“ histogramu na základe Scottovho pravidla

$$3.5std(data)\sqrt[3]{N}$$

Výpočet entropie

- ▶ Nepresnosti v reálnych fotkách – vstupný obrázok vyhladíme Gaussianom
- ▶ Vytvoríme histogram šedoúrovňového obrázku, vezmeme iba 90% stredných dát
- ▶ „Bin width“ histogramu na základe Scottovho pravidla

$$3.5std(data)\sqrt[3]{N}$$

- ▶ po normalizácii histogramu a vylúčení malých hodnôt získame entropiu

$$\eta = - \sum_i p_i \log p_i$$

p_i – veľkosť „binov“ histogramu

Získanie intrinsic obrázku – výsledný algoritmus

- ▶ Získaj 2D logaritmický chromatický obrázok

Získanie intrinsic obrázku – výsledný algoritmus

- ▶ Získaj 2D logaritmický chromatický obrázok
- ▶ for $\theta = 1..180$

Získanie intrinsic obrázku – výsledný algoritmus

- ▶ Získaj 2D logaritmický chromatický obrázok
- ▶ for $\theta = 1..180$
 - ▶ Získaj šedoúrovňový obrázok \mathcal{I} na základe projekcie bodov pod uhlom θ

Získanie intrinsic obrázku – výsledný algoritmus

- ▶ Získaj 2D logaritmický chromatický obrázok
- ▶ for $\theta = 1..180$
 - ▶ Získaj šedoúrovňový obrázok \mathcal{I} na základe projekcie bodov pod uhlom θ
 - ▶ Vypočítaj entropiu \mathcal{I}

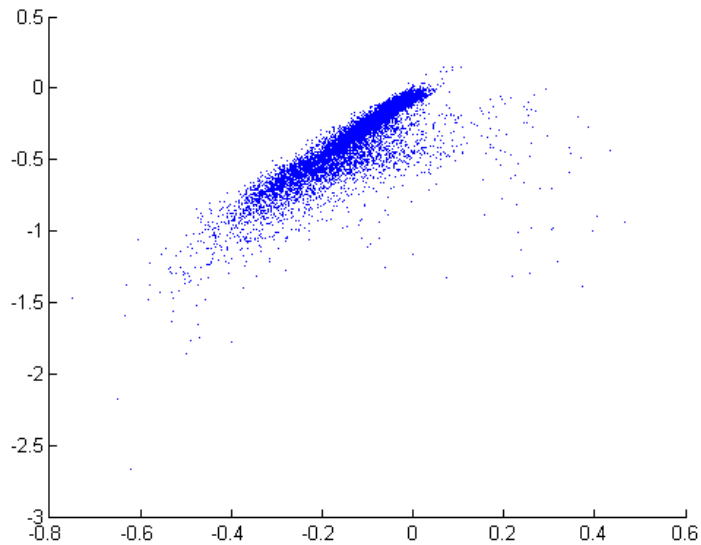
Získanie intrinsic obrázku – výsledný algoritmus

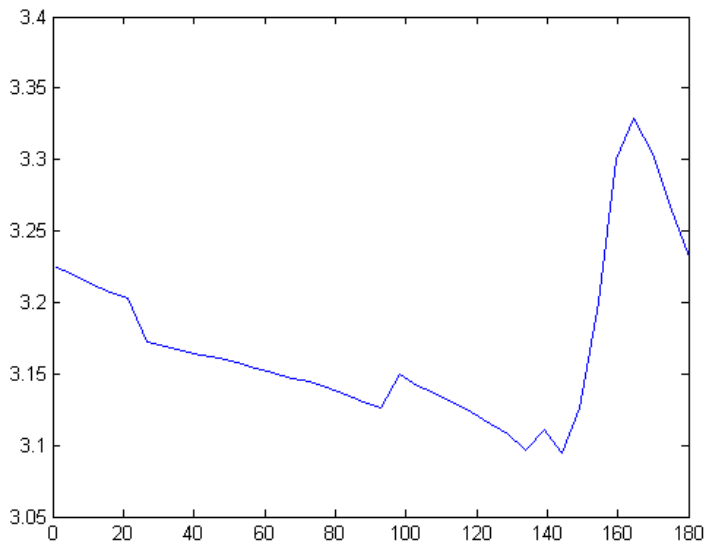
- ▶ Získaj 2D logaritmický chromatický obrázok
- ▶ for $\theta = 1..180$
 - ▶ Získaj šedoúrovňový obrázok \mathcal{I} na základe projekcie bodov pod uhlom θ
 - ▶ Vypočítaj entropiu \mathcal{I}
 - ▶ Ak je entropia nižšia ako doterajšie minimum, θ je najlepší odhad

Získanie intrinsic obrázku – výsledný algoritmus

- ▶ Získaj 2D logaritmický chromatický obrázok
- ▶ for $\theta = 1..180$
 - ▶ Získaj šedoúrovňový obrázok \mathcal{I} na základe projekcie bodov pod uhlom θ
 - ▶ Vypočítaj entropiu \mathcal{I}
 - ▶ Ak je entropia nižšia ako doterajšie minimum, θ je najlepší odhad
- ▶ Na základe najlepšej θ zrekonštruujeme šedoúrovňový intrinsic obrázok, pridáme energiu tak, aby medián 1% najjasnejších pixelov mal chromaticitu pôvodného obrázku









Identifikácia tieňa

- ▶ Ako získať tieň?

Identifikácia tieňa

- ▶ Ako získať tieň?
- ▶ Intrinsic — $X * \text{šedoúrovňový obrázok}$

Identifikácia tieňa

- ▶ Ako získať tieň?
- ▶ Intrinsic — $X * \text{šedoúrovňový obrázok}$
- ▶ Vyhľadanie masky, odstránenie šumu a dier pomocou morfológie

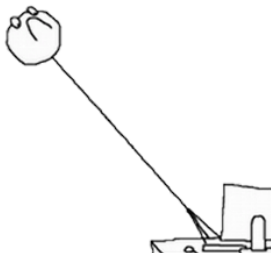
Identifikácia tieňa

- ▶ Ako získať tieň?
- ▶ Intrinsic – X * šedoúrovňový obrázok
- ▶ Vyhľadanie masky, odstránenie šumu a dier pomocou morfológie



Odstránenie tieňa

- Zvolený iný postup ako v pôvodnej práci



1. Initialization, $t = 0$, calculate:

$$\begin{aligned}(\nabla_x \rho'_k(x, y))^t &\rightarrow T_S(\nabla_x \rho'_k(x, y), q_s(x, y)) \\ (\nabla_y \rho'_k(x, y))^t &\rightarrow T_S(\nabla_y \rho'_k(x, y), q_s(x, y)).\end{aligned}$$

2. Update shadow edge pixels (i, j) :

$$\begin{aligned}(\nabla_x \rho'_k(i, j))^t &\rightarrow \\ (\nabla_x \rho'_k(i-1, j))^{t-1} &+ (\nabla_x \rho'_k(i, j-1))^{t-1} \\ (\nabla_x \rho'_k(i+1, j))^{t-1} &+ (\nabla_x \rho'_k(i, j+1))^{t-1},\end{aligned}$$

$$\begin{aligned}(\nabla_y \rho'_k(i, j))^t &\rightarrow \\ (\nabla_y \rho'_k(i-1, j))^{t-1} &+ (\nabla_y \rho'_k(i, j-1))^{t-1} \\ &+ (\nabla_y \rho'_k(i+1, j))^{t-1} + (\nabla_y \rho'_k(i, j+1))^{t-1}.\end{aligned}$$

3. Enforce integrability by projection onto integrable edge map [43], and integrate:

$$F_x(u, v) = \mathcal{F}[\nabla_x \rho'_k], \quad F_y(u, v) = \mathcal{F}[\nabla_y \rho'_k],$$

$$a_x = e^{2\pi i u/N} - 1, \quad a_y = e^{2\pi i v/M} - 1,$$

$$Z(u, v) = \frac{a_x^* F_x(u, v) + a_y^* F_y(u, v)}{|a_x|^2 + |a_y|^2}, \quad \rho'(0, 0) = 0,$$

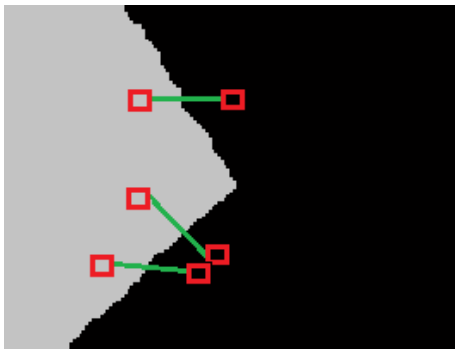
$$(\nabla_x \rho')^t = \mathcal{F}^{-1}[a_x Z], \quad (\nabla_y \rho')^t = \mathcal{F}^{-1}[a_y Z],$$

Odstránenie tieňa (zjednodušená verzia)

- Nájdeme obrysy tieňa a dotyčnice k nemu

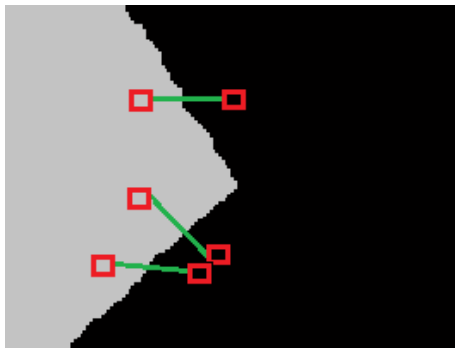
Odstránenie tieňa (zjednodušená verzia)

- ▶ Nájdeme obrysy tieňa a dotyčnice k nemu
- ▶ Na základe dotyčníc nájdeme pixely z oboch strán obrysov



Odstránenie tieňa (zjednodušená verzia)

- ▶ Nájdeme obrysy tieňa a dotyčnice k nemu
- ▶ Na základe dotyčníc nájdeme pixely z oboch strán obrysov
- ▶ Časť obrazu v tieni pre násobíme pomerom priemernej svetlosti pixelov vo svetle a tieni



Odstránenie tieňa



Odstránenie tieňa

- Problém pri nehomogénnom povrchu



Odstránenie tieňa

- ▶ Problém pri nehomogénnom povrchu
- ▶ Škaredé okraje



Odstránenie tieňa

- ▶ Problém pri nehomogénnom povrchu
- ▶ Škaredé okraje
- ▶ Pokus o ich vyhladenie vyprodukoval ešte škaredšie okraje



Odstránenie tieňa

- ▶ Pokus o zložitejší prístup pomocou watershed segmentácie

Odstránenie tieňa

- ▶ Pokus o zložitejší prístup pomocou watershed segmentácie
- ▶ Postupne zosvetliť oblasti v tieni od najsvetlejších okrajov smerom dovnútra

Odstránenie tieňa

- ▶ Pokus o zložitejší prístup pomocou watershed segmentácie
- ▶ Postupne zosvetliť oblasti v tieni od najsvetlejších okrajov smerom dovnútra
- ▶ Horšie výsledky než predošlá metóda

Ukážka & Otázky

