Noise Reduction in Photometric Stereo with Non-distant Light Sources

R. Kozera[1] and L. Noakes[2]

[1]School of Comp. Sc. and Software Engineering
[2]School of Mathematics and Statistics
The University of Western Australia
Perth, Australia

1. Shape from Shading

Shape from Shading: surface $\mathbf{S}$ from **image intensity** $\mathbf{E}$.

Sfs is modeled by **image irradiance equation** (**IIE**)

$$\boxed{\mathbf{R}(u_x, u_y) = \mathbf{E}(x, y)\ .}$$

Reflectance map: reflectance properties of $\mathbf{S}$.

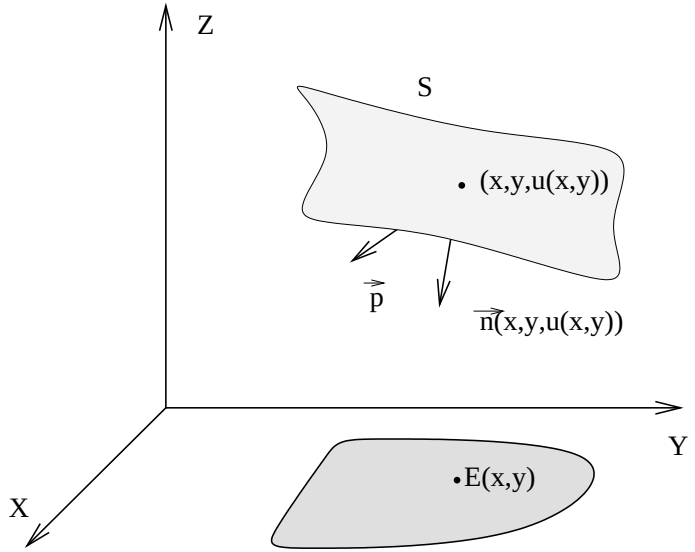


Fig. 1. Light direction $\vec{\mathbf{p}}$ **fixed** for $\mathbf{w} \in \mathbf{S}$.

Lambertian surface:

$$\frac{p_1 u_x(x,y) + p_2 u_y(x,y) - p_3}{\sqrt{p_1^2 + p_2^2 + p_3^2}\sqrt{u_x^2(x,y) + u_y^2(x,y) + 1}} = \mathbf{E}(x,y) \ .$$

1st order nonlinear PDE with $v = u + const$ ambiguity.

Lambertian surface: ideal matte surface.

Barium sulfate, magnesium carbonate, paper, sand, or matte paint.

Four important issues for IIE:

- **existence**
- **uniqueness**
- **construction of explicit sfs scheme**
- **real/noisy/digitized data**

2. Photometric Stereo: Distant Light Sources

Photometric Stereo - multiple **light-sources**.

Monochromatic (**polarized/color**) light: sequential (parallel) ill.

$$\begin{aligned}
\frac{p_1 u_x(x,y) + p_2 u_y(x,y) - p_3}{\sqrt{p_1^2 + p_2^2 + p_3^2}\sqrt{u_x^2(x,y) + u_y^2(x,y) + 1}} &= \mathbf{E_1}(x,y)\ , \\
\frac{q_1 u_x(x,y) + q_2 u_y(x,y) - q_3}{\sqrt{q_1^2 + q_2^2 + q_3^2}\sqrt{u_x^2(x,y) + u_y^2(x,y) + 1}} &= \mathbf{E_2}(x,y)\ , \qquad (1) \\
\frac{r_1 u_x(x,y) + r_2 u_y(x,y) - r_3}{\sqrt{r_1^2 + r_2^2 + r_3^2}\sqrt{u_x^2(x,y) + u_y^2(x,y) + 1}} &= \mathbf{E_3}(x,y)\ .
\end{aligned}$$

(1) is **well-posed** over $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$.

(i) Gradient computation:

Th. 1. For **Photometric Stereo** with **s=3**:

$$u_x = f_1(\mathbf{E_1}(x,y), \mathbf{E_2}(x,y), \mathbf{E_3}(x,y), \vec{\mathbf{p}}, \vec{\mathbf{q}}, \vec{\mathbf{r}}) \ ,$$

$$u_y = f_2(\mathbf{E_1}(x,y), \mathbf{E_2}(x,y), \mathbf{E_3}(x,y), \vec{\mathbf{p}}, \vec{\mathbf{q}}, \vec{\mathbf{r}}) \ .$$

(ii) Gradient integration:

u (thus $\mathbf{S} = graph(u)$) is:

$$\boxed{u(x,y) = u(x_0,y_0) + \int_\gamma u_x dx + u_y dy \ .}$$

$\gamma \subset \Omega$ any C^1 curve joining $(x,y) \in \Omega$ & fixed $(x_0,y_0) \in \Omega$.

Images with **noise**: (v_1, v_2) is **not integrable** - **no** $u \in C^1(\Omega)$

$$\boxed{u_x = v_1 \quad \text{and} \quad u_y = v_2 \, .}$$

For Ω **simply connected**, $\vec{\mathbf{v}}$ is **integrable**:

a) if for $\mathbf{v} \in C^0$:

$$\boxed{\int_{\gamma_c} v_1(x,y)dx + v_2(x,y)dy = 0 \, ,}$$

for each piecewise smooth **closed loop** $\gamma_s \subset \Omega$.

b) if for $\mathbf{v} \in C^1$:

$$\boxed{\frac{\partial v_1}{\partial y}(x,y) = \frac{\partial v_2}{\partial x}(x,y) \, .}$$

Add **Gaussian Noise** (fixed **mean** & **std**) to images.

Discrete integrability for **atomic closed loop** γ_c^a:

$$\boxed{\Delta x v_x^f[i,j] + \Delta y v_y^f[i+1,j] - \Delta x v_x^f[i,j+1] - \Delta y v_y^f[i,j] = 0\ ,}$$

$$v_x^f[i,j] = \frac{u_{i+1}^j - u_i^j}{\Delta x}\ , \qquad v_x^f[i,j+1] = \frac{u_{i+1}^{j+1} - u_i^{j+1}}{\Delta x}\ ,$$

$$v_y^f[i,j] = \frac{u_i^{j+1} - u_i^j}{\Delta y}\ , \qquad v_y^f[i+1,j] = \frac{u_{i+1}^{j+1} - u_{i+1}^j}{\Delta y}\ .$$

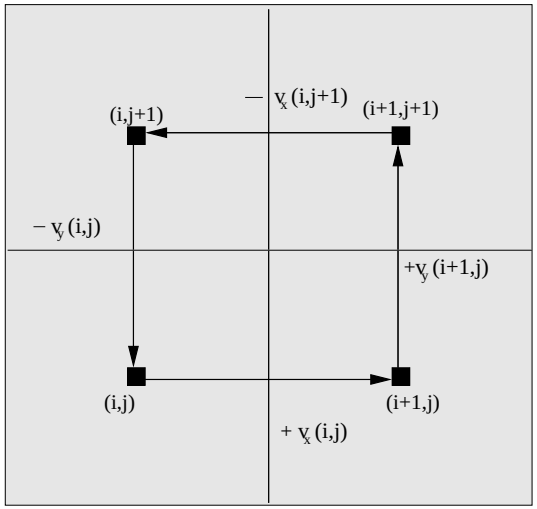


Fig. 2. Discrete integrability for atomic loop γ_c^a.

Discrete integrable fields $X \in V_I \subset \mathbb{R}^m$ (m big)

$$\boxed{L(X) = \vec{\mathbf{0}}} \quad \textbf{due to noise} \quad \boxed{L(\vec{\mathbf{v}}) \neq \vec{\mathbf{0}}} \quad \vec{\mathbf{v}} \notin V_I \ .$$

Maximum likelihood estimate $\hat{\mathbf{v}}_{opt} \in V_I$ to $\vec{\mathbf{v}}$:

$$\boxed{\mathcal{J}_Q(\hat{\mathbf{v}}_{opt.}) = \min_{\hat{\mathbf{w}} \in V_I} \sum_{i,j=1} \Big((\hat{w}_1[i,j] - v_1[i,j])^2 + (\hat{w}_2[i,j] - v_2[i,j])^2 \Big) \ .}$$

$\mathcal{J}_Q$ is **quadratic** & **strictly convex**: one global $\hat{\mathbf{v}}_{opt} \in V_I$:

$$\hat{\mathbf{v}} = \vec{\mathbf{v}} - L^T (L L^T)^{-1} (L(\vec{\mathbf{v}})) \ .$$

Inversion problem $(LL^T)^{-1}$ (big size).

Iterative (Gauss-Seidel) or **sparse matrix** schemes are feasible.

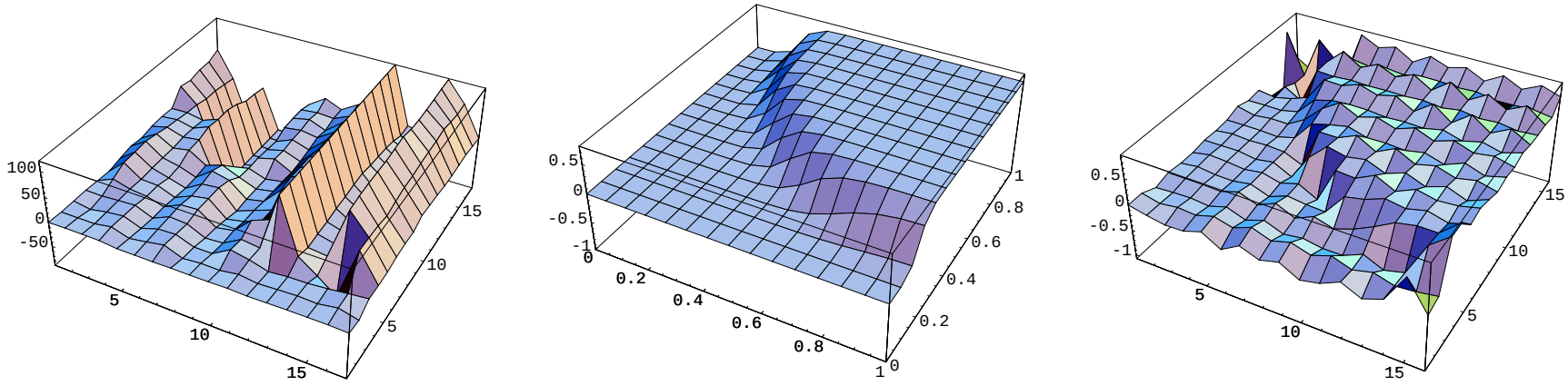
Denoising Linear method is **flawed**.

"**Nonlinear filter**" in **Th. 1**

$$u_x + \mathcal{N}(0,\lambda) \neq v_1 = f_1(\mathbf{E_1} + \mathcal{N}(0,\sigma), \mathbf{E_2} + \mathcal{N}(0,\sigma), \mathbf{E_3} + \mathcal{N}(0,\sigma), \vec{\mathbf{p}}, \vec{\mathbf{q}}, \vec{\mathbf{r}}) \ ,$$

$$u_y + \mathcal{N}(0,\lambda) \neq v_2 = f_2(\mathbf{E_1} + \mathcal{N}(0,\sigma), \mathbf{E_2} + \mathcal{N}(0,\sigma), \mathbf{E_3} + \mathcal{N}(0,\sigma), \vec{\mathbf{p}}, \vec{\mathbf{q}}, \vec{\mathbf{r}}) \ .$$

does **not retain Gaussian noise**.



(a) **(b)** **(c)**

Fig. 3. a) Linear. **b)** Noiseless. **c)** Nonlinear Leap-Frog.

For images of **tangensoidal surface** $\sigma = 0.25$.

3. Non-distant Light Sources

Set non-distant **light-source** at $\mathbf{p} = (p_1, p_2, p_3)$.

Unit light direction $\vec{\mathbf{p}}(x, y, u(x, y)) = \frac{(p_1-x, p_2-y, p_3-u(x,y))}{\sqrt{(p_1-x)^2+(p_2-y)^2+(p_3-u(x,y))^2}}$.

IEE at $(x, y, u(x, y)) \in S$:

$$\frac{(p_1-x)u_x(x,y)+(p_2-y)u_y(x,y)-(p_3-u(x,y))}{\sqrt{(p_1-x)^2+(p_2-y)^2+(p_3-u(x,y))^2}\sqrt{u_x^2(x,y)+u_y^2(x,y)+1}} = \mathbf{E}(x, y) \ .$$

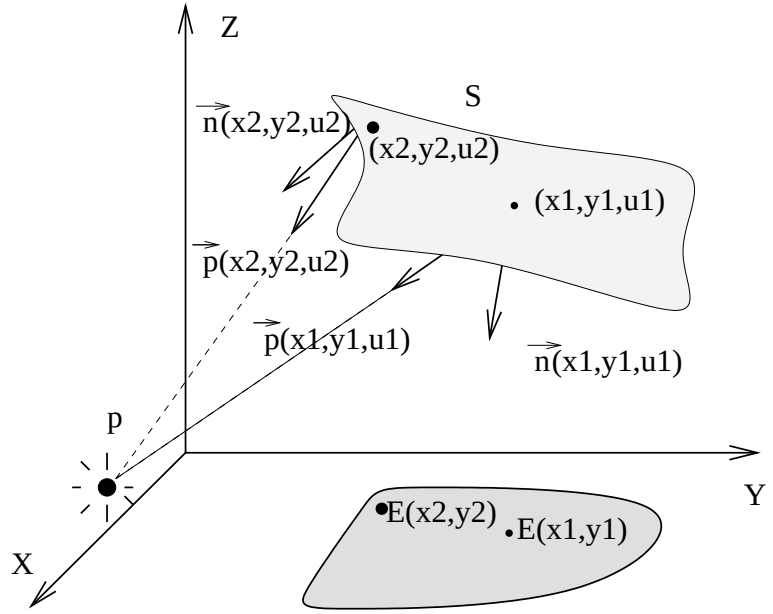


Fig. 4. Light direction $\vec{\mathbf{p}}(x, y, u(x, y))$ **varies** with $\mathbf{w} \in \mathbf{S}$.

Gradient formulae from Th. 1:

$$u_x(x,y) = g_1(\boxed{u}, x, y, \mathbf{p}, \mathbf{q}, \mathbf{r}) \ , \quad u_y(x,y) = g_2(\boxed{u}, x, y, \mathbf{p}, \mathbf{q}, \mathbf{r}) \ .$$

For **noisy** $\widehat{\mathbf{v}}(u)$ we **need**:

- **correct cost function** $\mathcal{J}$ (both (non)distant light sources).
- **feasible scheme** to optimize $\mathcal{J}$.

2D Leap-Frog: for **optimization** with **big number of parameters**.

Noisy P. Stereo & **2D LF** for distant sources was analyzed.

Discretization: central-diff. (u_x, u_y) approximations ($\Omega = [0,1]^2$ and pixel size $1/M$) yield at each pixel **IIE**

$$\boxed{f_{ij}^p =< \widehat{\nu}_i^j, p^{ij} >= E_i^j \ ,} \qquad 1 \le i, j \le M$$

$$\begin{aligned}
\widehat{\nu}_i^j &= \frac{(M\frac{(u_{i+1}^j - u_{i-1}^j)}{2}, M\frac{(u_i^{j+1} - u_i^{j-1})}{2}, -1)}{\sqrt{1 + (M\frac{(u_{i+1}^j - u_{i-1}^j)}{2})^2 + (M\frac{(u_i^{j+1} - u_i^{j-1})}{2})^2}} \ , \\
p^{ij} &= (p_1 - x_i, p_2 - y_j, p_3 - u_i^j) \ .
\end{aligned}$$

$\vec{u} \in V = \mathbb{R}^{M^2-4}$ is stored in the $M \times M$ **tableau**:

$$\vec{u} = \begin{pmatrix}
 & u_2^M & \cdots & \cdots & u_{M-1}^M & \\
u_1^{M-1} & u_2^{M-1} & \cdots & \cdots & u_{M-1}^{M-1} & u_M^{M-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
u_1^2 & u_2^2 & \cdots & \cdots & u_{M-1}^2 & u_M^2 \\
 & u_2^1 & \cdots & \cdots & u_{M-1}^1 &
\end{pmatrix} .$$

For **internal pixels** $f^p : V = \mathbb{R}^{M^2-4} \to \mathbb{R}^{(M-2)^2}$

$$\boxed{f^p(\vec{u}) = E\ .}$$

Add **Gaussian noise** $\mathcal{N}(0, \sigma)$ to images. Max. likelihood est. $\vec{u}^{opt} \approx \vec{u}$:

$$\boxed{\mathcal{J}(\vec{u}^{opt}) = \min_{\vec{u}\in V} \Sigma_{s=1}^{3} \|f^{p^s}(\vec{u}) - E^s\|^2\ .} \qquad (2)$$

$\mathcal{J}$ is **nonlinear optimization** (finds $\vec{u}^{opt}$ directly).

Numerical methods for (2):

- **Newton's method** for $\boxed{\nabla\mathcal{J} = \vec{0}}$: inversion of big size $D^2\mathcal{J}$ ($\approx (600 \times 800)^2$) with each local iteration.

- **Gradient Descent**: each step-size needs eigenvalues of $D^2\mathcal{J}$.

4. 2D Leap-Frog Optimization Scheme

Feasible option: **a nonlinear 2D Leap-Frog**:

$$\boxed{\vec{u}_n = LF(\vec{u}_{n+1})} \quad \text{iteration of iterations}$$

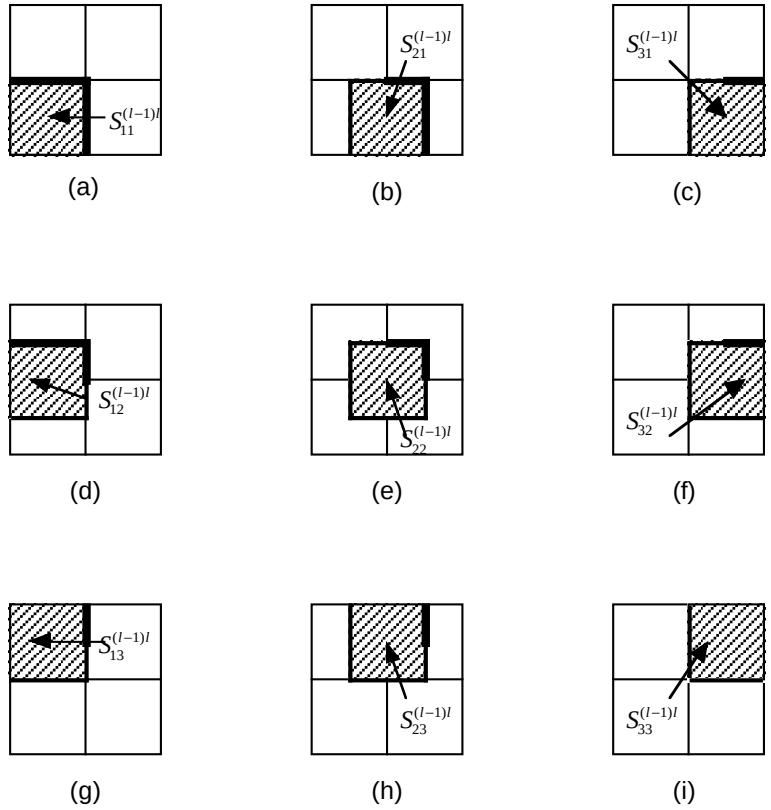
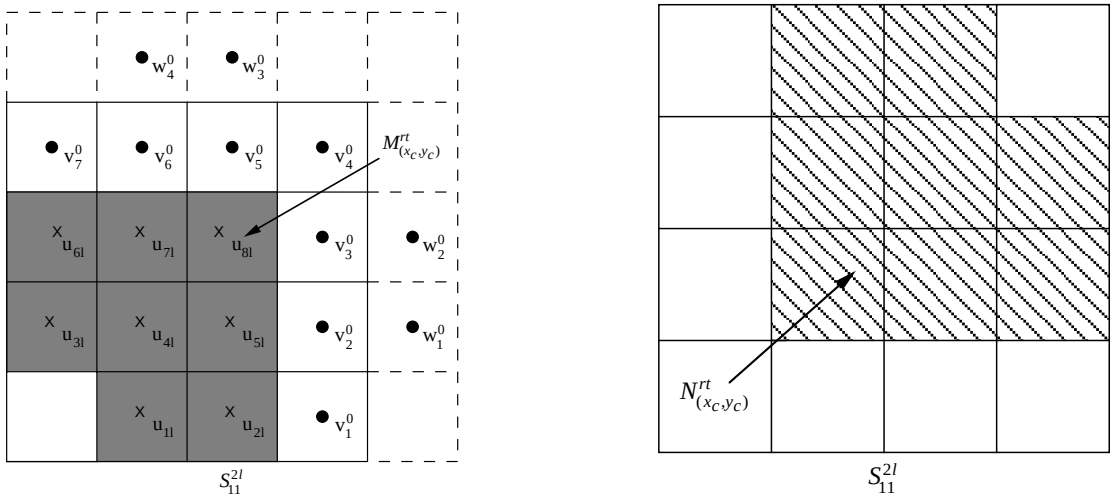


Fig. 5. One **2D-LF** iteration: snapshot optimizations.

- **initial guess** $\vec{u}_0$ needed.
- **snapshot optima update** local variables of $\vec{u}_n$.

$$\mathcal{J}(\vec{u}_n^{+}) = \mathcal{J}_0(\vec{u}_n^{fix}) + \mathcal{J}_1(\vec{u}_n^{+opt.}) \leq \mathcal{J}_0(\vec{u}_n^{fix}) + \mathcal{J}_1(\vec{u}_n^{-opt.}) = \mathcal{J}(\vec{u}_n^{-}) \; .$$

- **1 iter. overlaps** update all values in $\vec{u}_n$.
- snapshot **size** and **order** can vary.



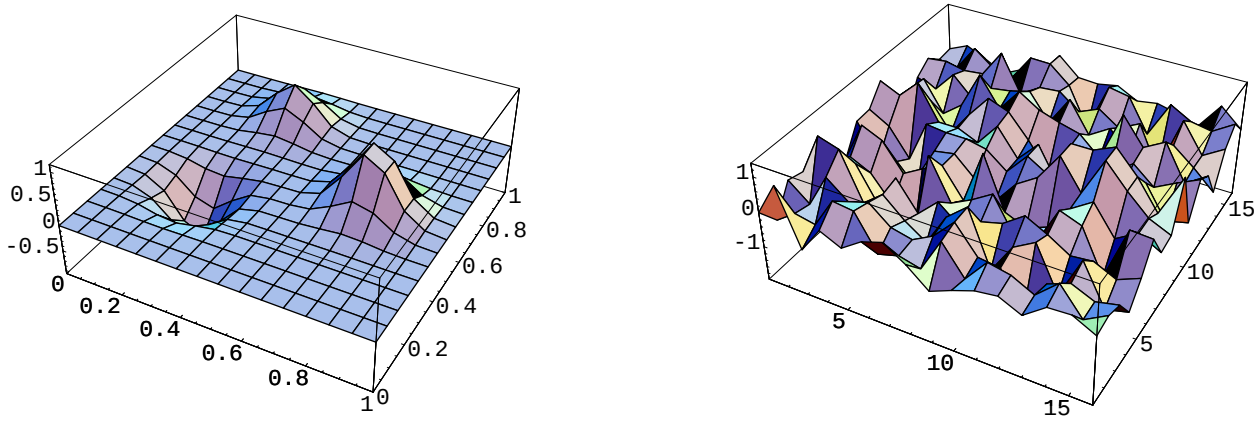
a(i) **a(ii)**

Fig. 6. i) Free variables: black pixels. **ii)** $\mathcal{J}_0$ minimized at shaded pixels.

5. Experiments - Non-distant Light Sources

Mathematica FindMinimum optimizes snapshots (4×4 pixels).

Ex. 1 $M = 16$ (finer grid).

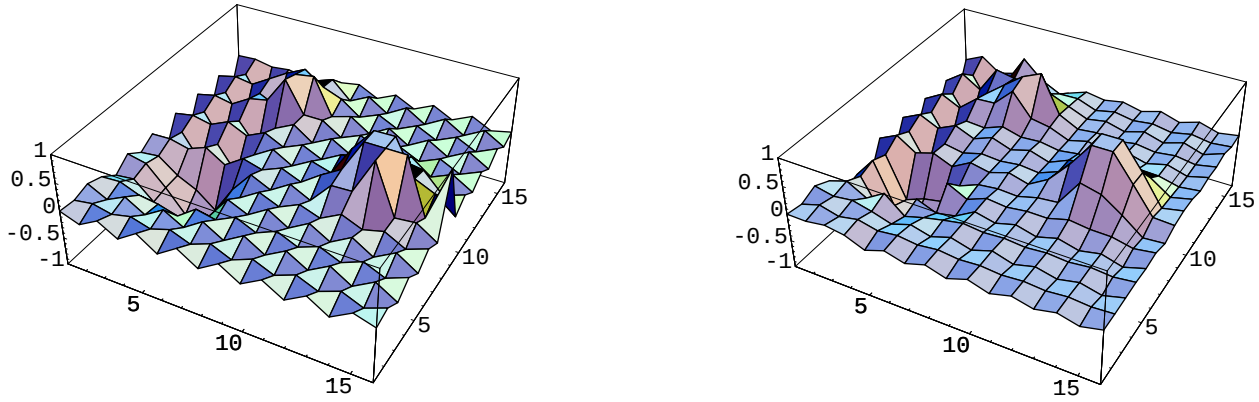


(a) **(b)**

Fig. 7. a) bumpy surface S_b. b) initial guess.

i) 3 sources $\mathcal{L}_1 = \{(0, 0, -8), (8, 8, -8), (5, 0, 11)\}$; $E_i + \mathcal{N}(0, 0.02)$.

ii) 3 sources $\mathcal{L}_2 = \{(0, 0, -60), (48, 48, -48), (32, 0, 64)\}$; $E_i + \mathcal{N}(0, 0.05)$.



(c) **(d)**

Fig. 8. S_b by 2D Leap-Frog **c)** for (i). **d)** for (ii).

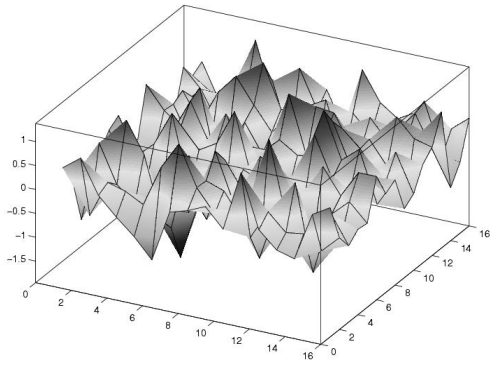
Initial values:

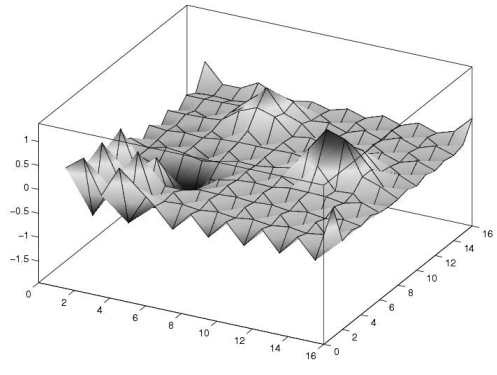
$$\mathcal{J}_{(i)}(\vec{u}_0) = 240.932\ , \qquad \mathcal{J}_{(ii)}(\vec{u}_0) = 247.088\ ,$$

dropped to

$$\mathcal{J}_{(i)}(\vec{u}_{110}) = 2.15855\ , \qquad \mathcal{J}_{(ii)}(\vec{u}_{110}) = 3.20951\ .$$

Mathematica: 110 iterations of **2D Leap-Frog** takes 12 hours.





Implementation in **C** is much **faster**:

Initial values ($M = 16$):

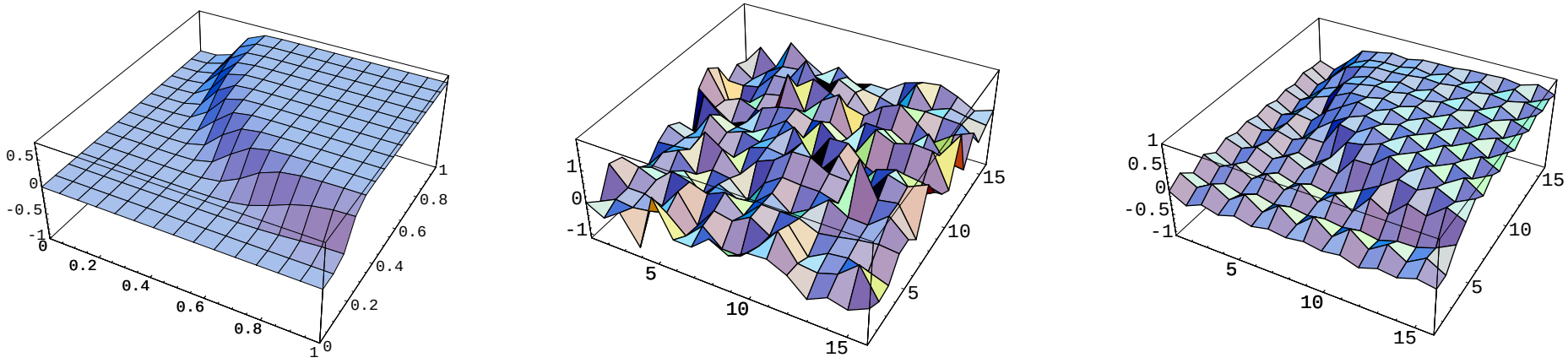
$$\mathcal{J}_{(ii)}(\vec{u}_0^{new}) = 242.219955$$

dropped to

$$\mathcal{J}_{(ii)}(\vec{u}_{100}) = 5.009533\,. \quad (\textbf{after } \mathbf{4.32sec.})$$

500 iter. (distant light source), image 256×256, Snapshot 4×4, Noise $\mathcal{N}(0, 0.02)$ 1 processor: $\approx$**11000 sec.** 8 processors: $\approx$**1300 sec.**.

Ex. 2 $M = 16$ (finer grid).



(a) **(b)** **(c)**

Fig. 9. a) tangensoidal. b) initial guess. c) 2D Leap-Frog

$\mathcal{L} = \{(0, 0, -2), (1.15, 1.15, -1.15), (0.89, 0, -1.79)\}$; $E_i + \mathcal{N}(0, 0.05)$.

Initial value

$$\mathcal{J}(\vec{u}_0) = 247.688$$

dropped (1 hour in Mathematica) to

$$\mathcal{J}(\vec{u}_{10}) = 1.42719\ .$$

5. Conclusions

- 2D Leap-Frog does well to recover shape from noisy data (at least in examples with uniform Gaussian noise added to 3 photographs).

- 2D Leap-Frog seems robust with respect to initial guess.

- 2D Leap-Frog is amenable to parallel computing schemes.

- 2D Leap-Frog can be used to other optimization schemes with large number of parameters (1 or 2 light-sources).

- open problems: finding good initial guess? Acceleration of LF?

Some Authors' Relevant References:

[1] Kozera R. (1992) "On shape recovery from two shading patterns". Int. J. Patt. Rec. Art. Int. **6**, (4):673–698.

[2] Kozera, R. (1991) "Existence and uniqueness in photometric stereo". Appl. Math. Comput. **44**, (1):1–104.

[3] Noakes L. (1999) "A global algorithm for geodesics". J. Math. Austral. Soc., **64**, 37–50.

[4] Noakes L. and Kozera R. (2001) "The 2-D Leap-Frog, noise, and digitization". in Geometry, Morphology and Computational Imaging". LNCS 2616, Springer-Verlag, 419–436.

[5] Noakes L. and Kozera R. (2002) "The 2-D Leap-Frog, noise, and digitization". in Digital and Image Geometry". LNCS 2242, Springer-Verlag, 352–364.

[6] Noakes L. and Kozera R. (2003) "Nonlinearities and noise reduction in 3-source photometric stereo". J. Math. Imag. Vision **18**: 119–127.