Automatic Illumination of Flat-Colored Drawings By 3D Augmentation of 2D Silhouettes

D. Tschumperlé, C. Porquet and A. Mahboubi

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, F-14050 Caen, France

ICIP'2022 - Bordeaux / France - 16-19 October 2022

GREYC Electronics and Computer Science Laboratory

Normandie Université

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

Context : Tools for Digital Illustration

A typical workflow for producing digital illustrations involves these 3 steps:

1) **Lineart creation**
2) Flat colorization
3) Illumination

← **Lineart =** gray-levels strokes on a transparent or white background

(lineart, courtesy of D. Revoy, from his webcomic "***Pepper & Carrot***")

Context : Tools for Digital Illustration

A typical workflow for producing digital illustrations involves these 3 steps:

1) Lineart creation
2) Flat colorization
3) Illumination

← **Flat Colorization =** additional background layer with constant colors

Context : Tools for Digital Illustration

A typical workflow for producing digital illustrations involves these 3 steps:

1) Lineart creation
2) Flat colorization
3) Illumination

← **Flat Colorization =** additional background layer
with constant colors

A Fast and Efficient Semi-guided Algorithm for Flat Coloring Linearts.
(S. Fourey, D. Tschumperlé, D. Revoy).
EUROGRAPHICS International Symposium on Vision, Modeling and Visualization 2018, Stuttgart, October 2018.

Context : Tools for Digital Illustration

A typical workflow for producing digital illustrations involves these 3 steps:

1) Lineart creation
2) Flat colorization
3) **Illumination**

← **Illumination =** additional foreground layer that adds light & shadows

Context : Tools for Digital Illustration

A typical workflow for producing digital illustrations involves these 3 steps:

1) Lineart creation
2) Flat colorization
3) **Illumination**

← **Illumination =** additional foreground layer that adds light & shadows

Our Goal: Automatic Creation of an Illumination Layer

Our Goal: Define an **algorithm** that:

- Automatically generates an illumination layer from an input 2D silhouette;
- Is computationally lightweight.

← **Input 2D silhouette**

← **Estimated illumination layer**

Proposed Method

Our algorithm has two steps:

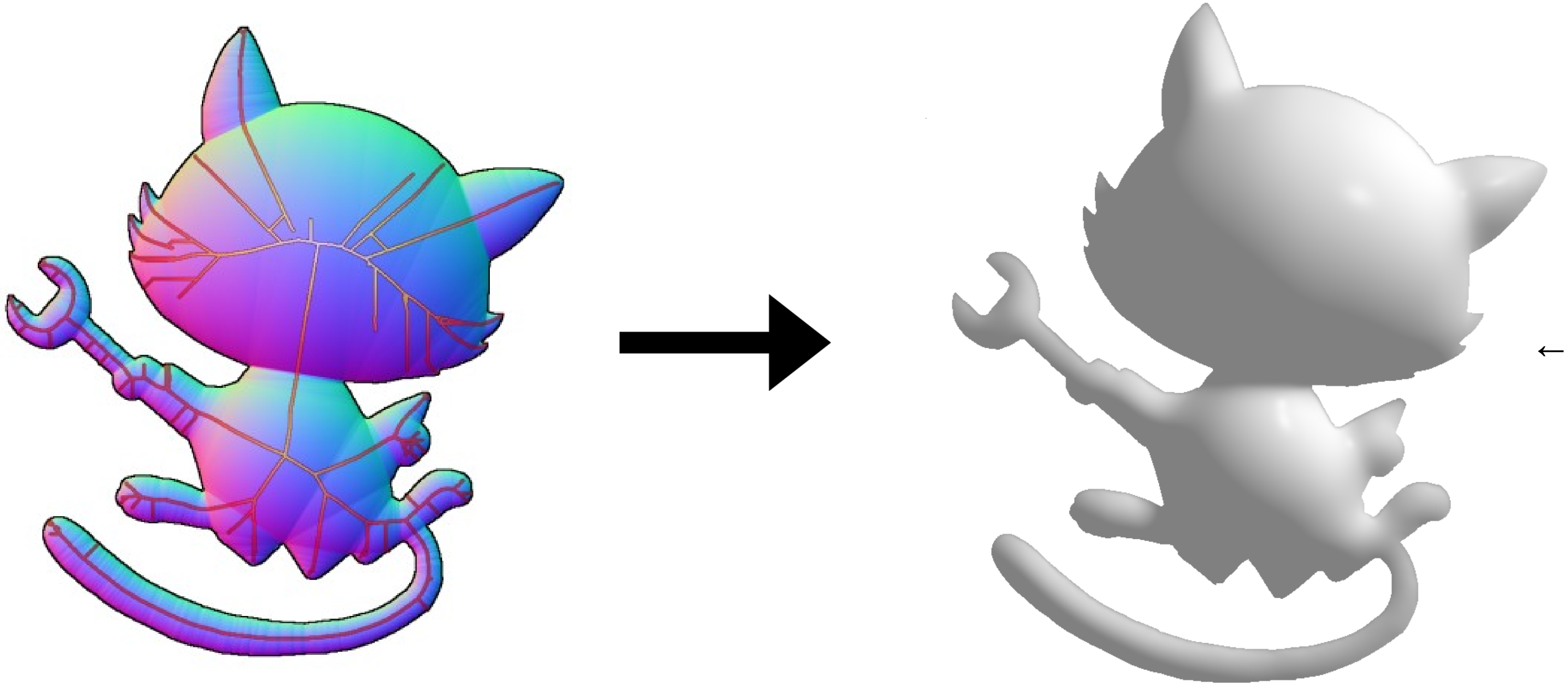
- **STEP 1: Estimate a plausible 3D elevation map from the 2D silhouette.**
- **STEP 2:** Create the illumination layer from the corresponding 3D normal map.

(*a*) Flat-colored drawing (*b*) Estimated 3D elevation map (*c*) Estimated normal map

Proposed Method

Our proposed algorithm has two steps:

- **STEP 1:** Estimate a plausible 3D elevation map from the 2D silhouette
- **STEP 2: Create the illumination layer from the corresponding 3D normal map.**

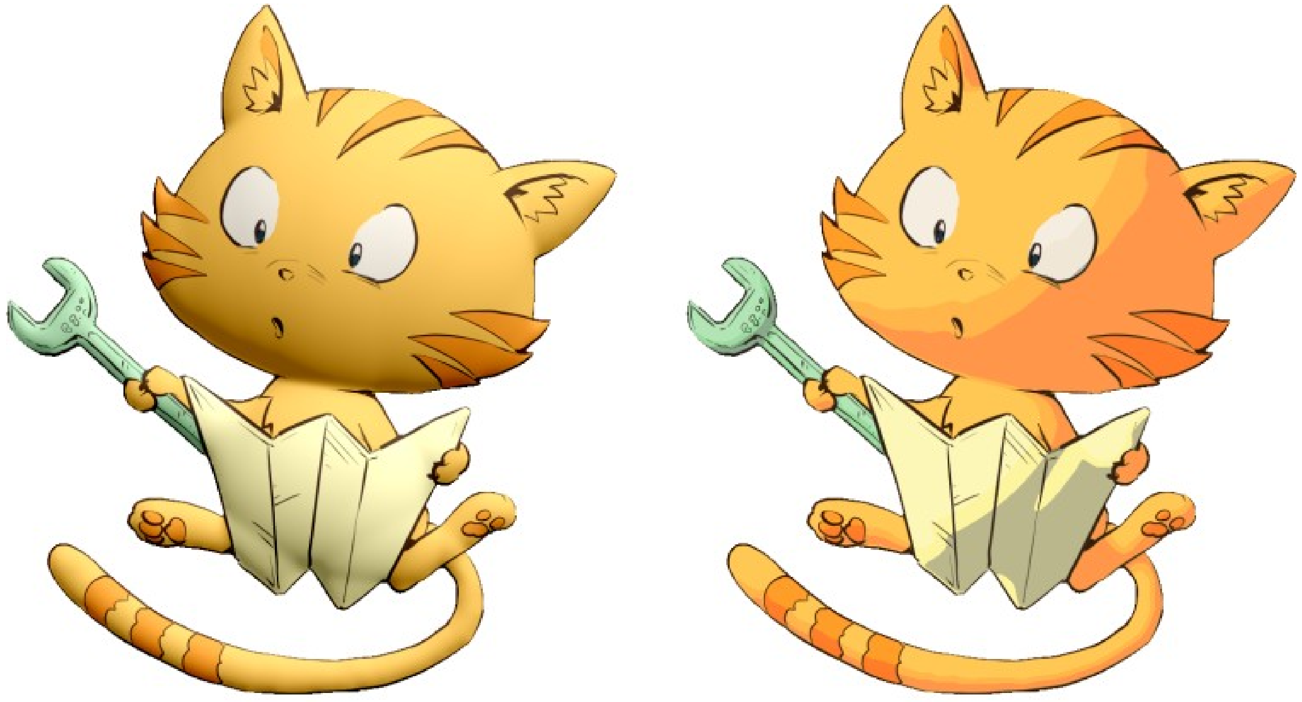


← **Estimated illumination layer**

Proposed Method

Our proposed algorithm has two steps:

- **STEP 1:** Estimate a plausible 3D elevation map from the 2D silhouette
- **STEP 2: Create the illumination layer from the corresponding 3D normal map.**



(d) Two examples of automatic illumination rendering

STEP 1 : Estimation of a 3D Elevation Map

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

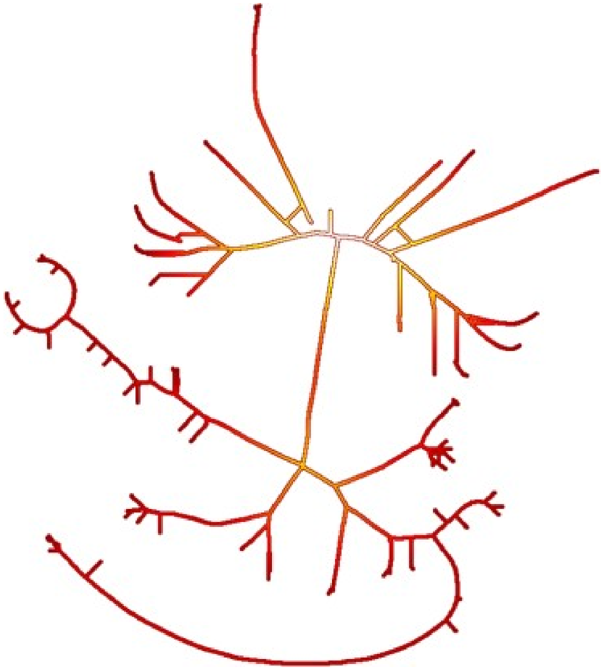
ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Estimation of a 3D Elevation Map

The 3D elevation map **B** (a.k.a. b*ump map*) is reconstructed from the skeleton **K** of the 2D silhouette **S**. (**K**, computed as the medial axis: Harry Blum (1967). "*A transformation for extracting new descriptors of shape*").

Input 2D silhouette **S**

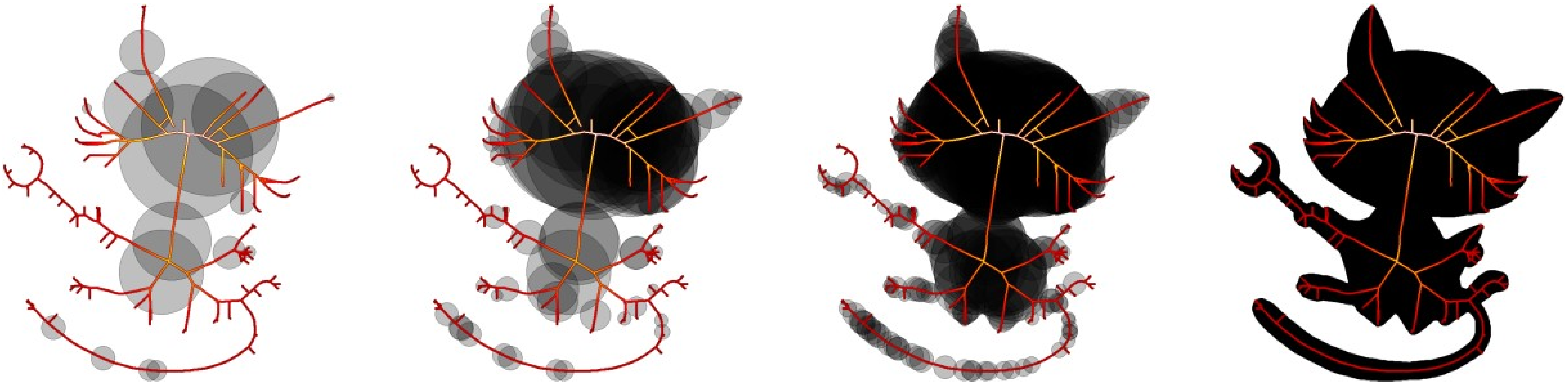


Skeleton **K**

Estimation of a 3D Elevation Map

Fact: It is possible to reconstruct the 2D binary silhouette **S** from its skeleton **K.**

- At each point **P=(x,y)** of **K**, draw a disk of radius **R(P)**.
- **R(P)** has to be the distance from **(x,y)** to the closest border of **S** (distance function).
- The intersection of all these disks gives **S** back.

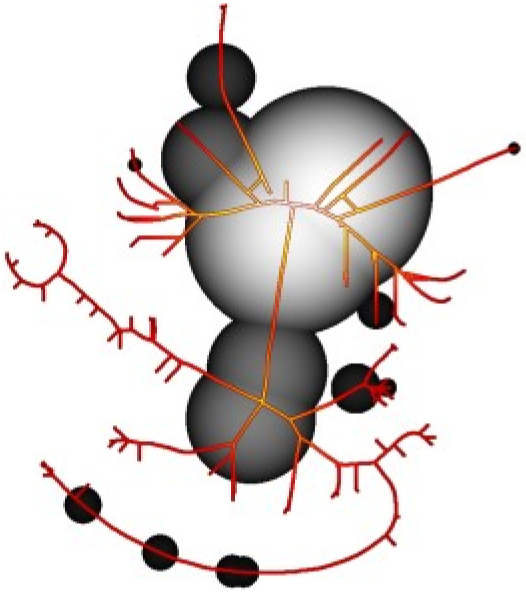


15 disks | 50 disks | 150 disks | All disks: Reconstructed silhouette **S**

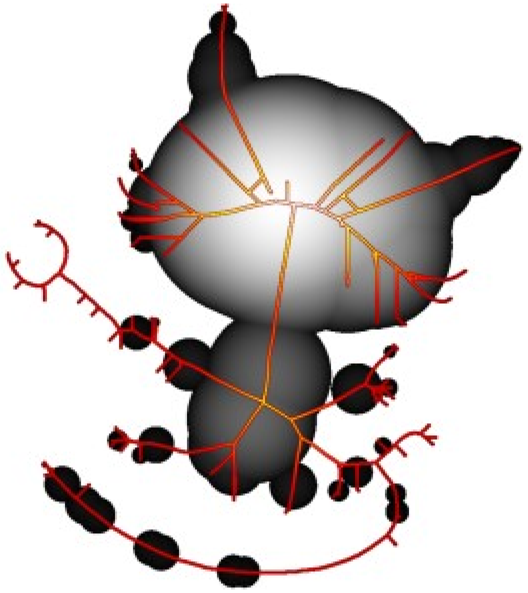
Estimation of a 3D Elevation Map

Idea: Rather than drawing 2D disks, we draw elevations of 3D semi-spheres instead:

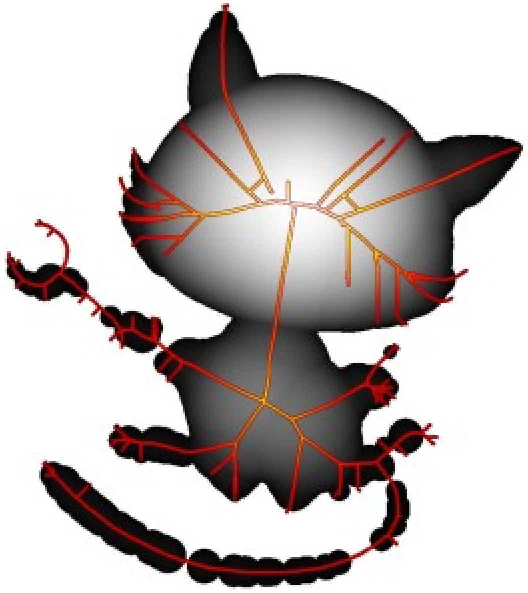
- At each point **(x,y)** of **K**, draw the 2D function that is the elevation of a semi-sphere of radius **R(P).**
- **R(P)** is chosen to be the distance from **(x,y)** to the closest border of **S** (distance function).
- The “intersection” (***max(…)***) of all these functions estimates a 3D elevation map **B**.



15 semi-spheres



50 semi-spheres



150 semi-spheres

All semi-spheres:
Estimated bump map **B**

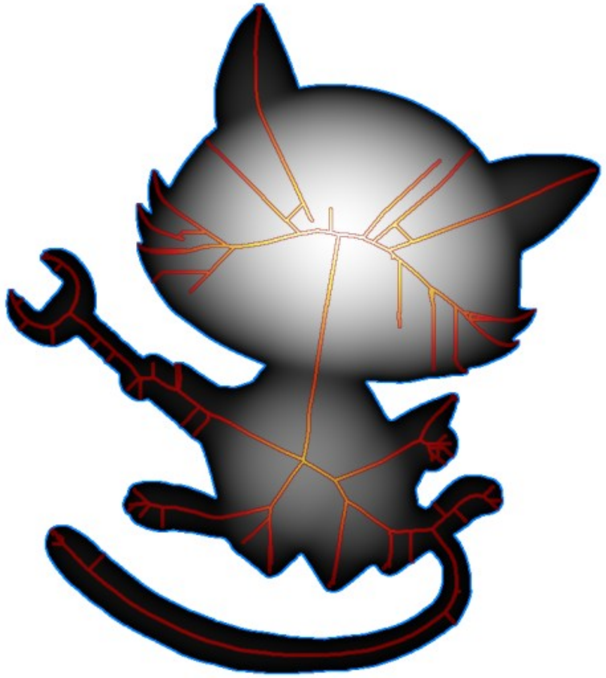
Estimation of a 3D Elevation Map

Algorithm in a nutshell:

$$B(P) = \max_{Q \in \mathcal{K}} \; g_Q(P), \quad \text{with} \quad g_Q(P) = \sqrt{\max(0, D(Q)^2 - \|P-Q\|^2)}$$

and $$D(P) = \min_{Q \in \Omega} \|P - Q\| \quad | \quad S(Q) = 0$$

2D silhouette **S**



Bump map **B**

STEP 2 : Estimation of the Illumination Layer

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Generation of the Illumination Layer

The illumination layer **J** is generated by applying the **Phong Illumination Model**
(B.T. Phong (1975), "*Illumination for computer generated pictures*"):

$$J(P) = K_a + K_d\,(\hat{L}(P) \cdot \hat{N}(P)) + K_s\;\max(0, R(P) \cdot \hat{V}(P))^{\alpha}$$

where

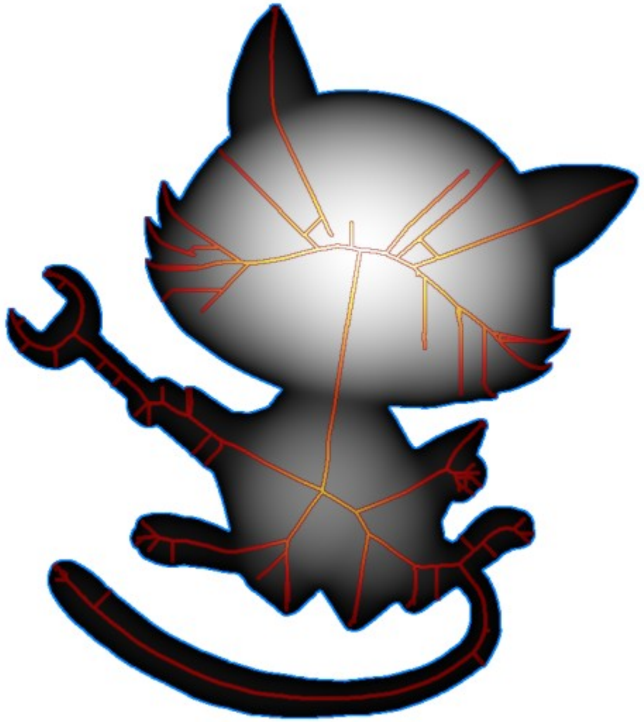
$$\begin{cases} L(P) & = & (L_x - P_x, L_y - P_y, L_z)^T \\ V(P) & = & (C_x - P_x, C_y - P_y, C_z)^T \\ R(P) & = & 2\,(\hat{N}(P) \cdot \hat{L}(P))\,\hat{N}(P) - \hat{L}(P) \end{cases}$$

→ Sum of an ambient light (**Ka**), a diffuse component (**Kd**) and a specular component (**Ks**).
→ Parameters of the model :
Light position **L = (Lx,Ly,Lz)**, Camera position **C = (Cx,Cy,Cz)**, Specularity coefficient **α**.
→ Require **N(P)**, 3D normal to any point **P** → Computation of the normal map **N** from the bump map **B**.

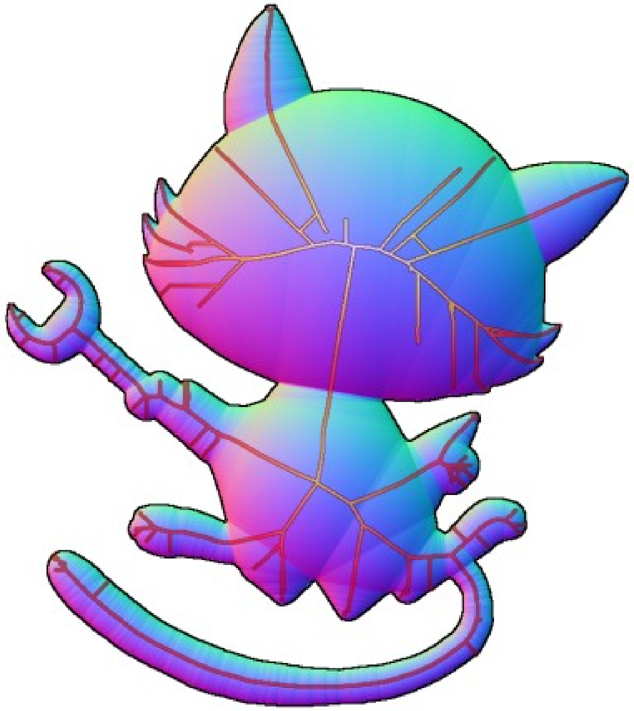
Generation of the Normal Map

Transformation of the bump map **B** into the normal map **N**:

For any point P, $$N(P) = \left(\frac{\partial B}{\partial x}(P), \frac{\partial B}{\partial y}(P), -1 \right)^T$$ (+ normalization to unit vector).



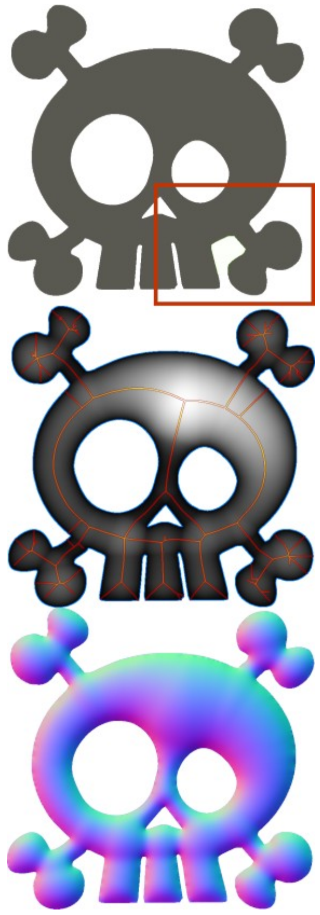
Bump map **B**

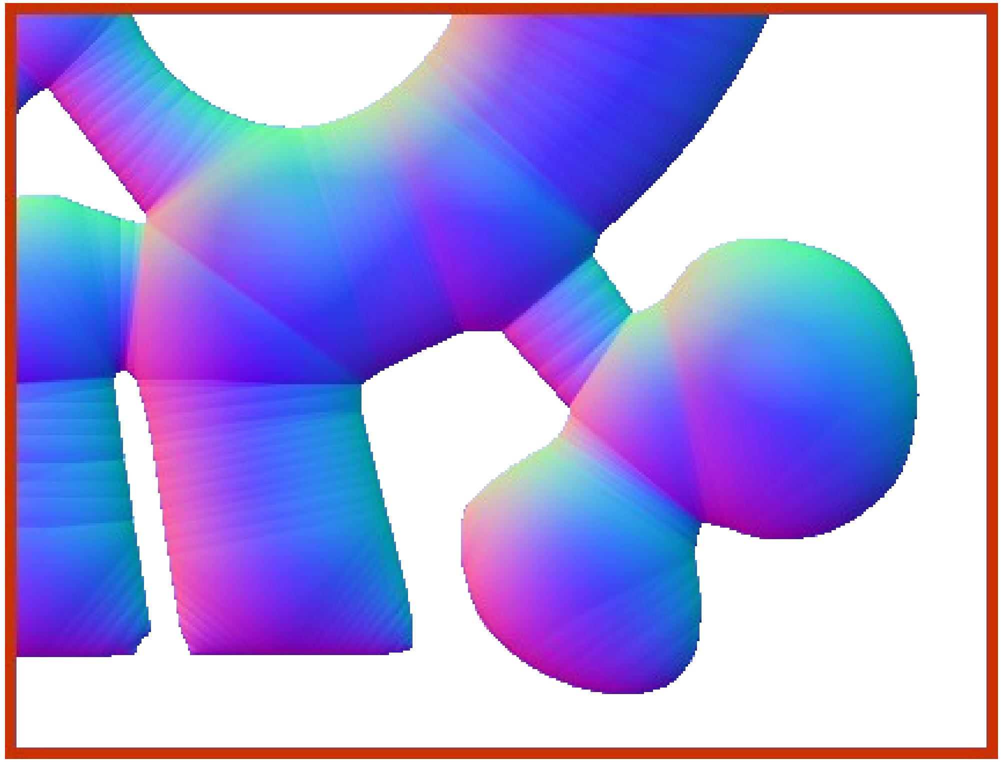


Normal map **N**

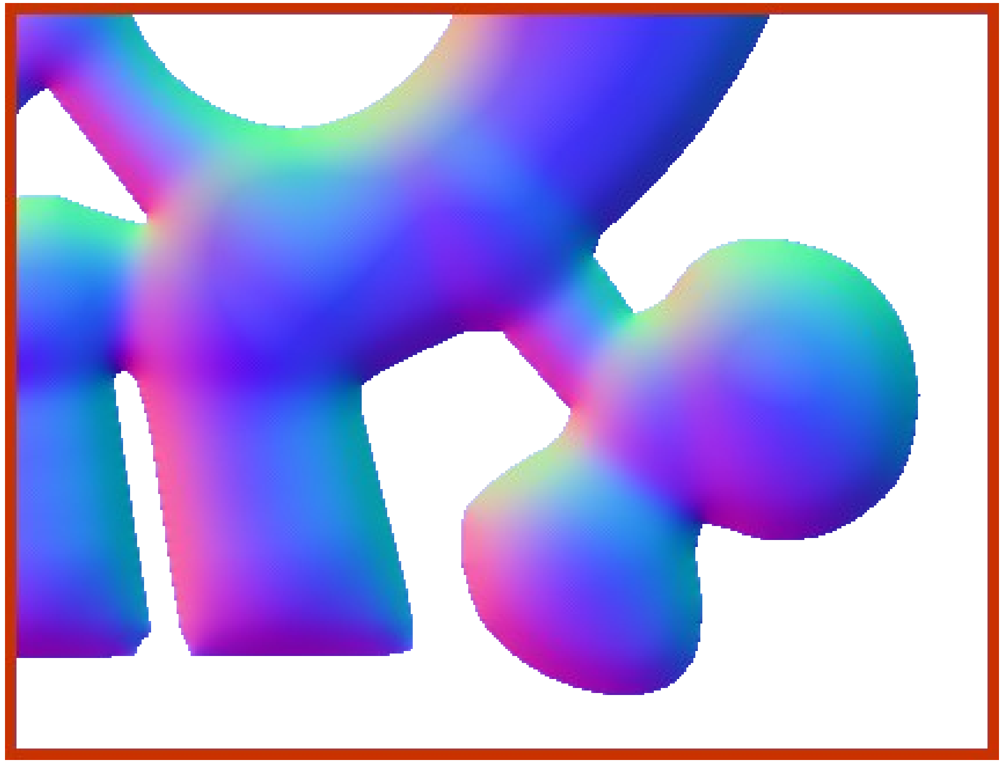
Normal Map Regularization

The fusion of semi-spheres may produces visible discontinuities in the normal map **N.**
→ **Guided filtering** (applied on map **N**) can be used to get rid of those, using the silhouette **S** as a guide.
(K. He, J. Sun, and X. Tang (2012), “*Guided image filtering*”).





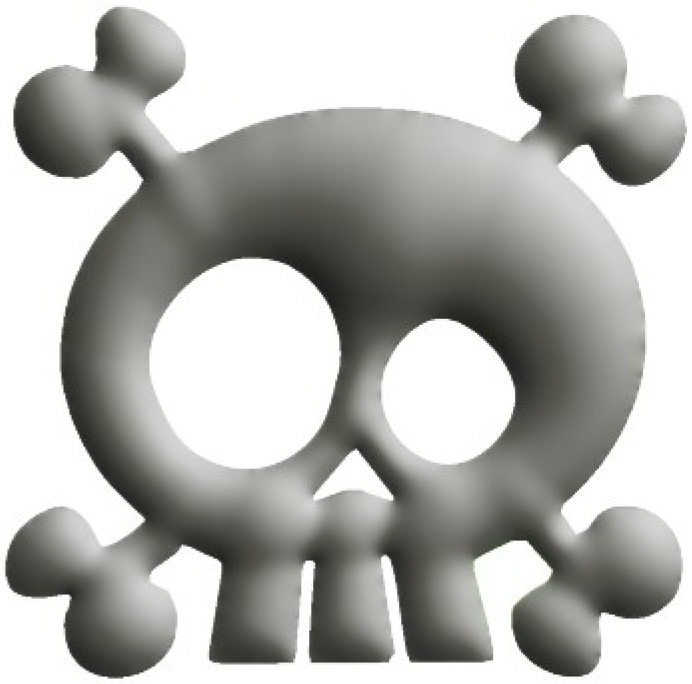
Normal map **N** *without regularization* (zoom)



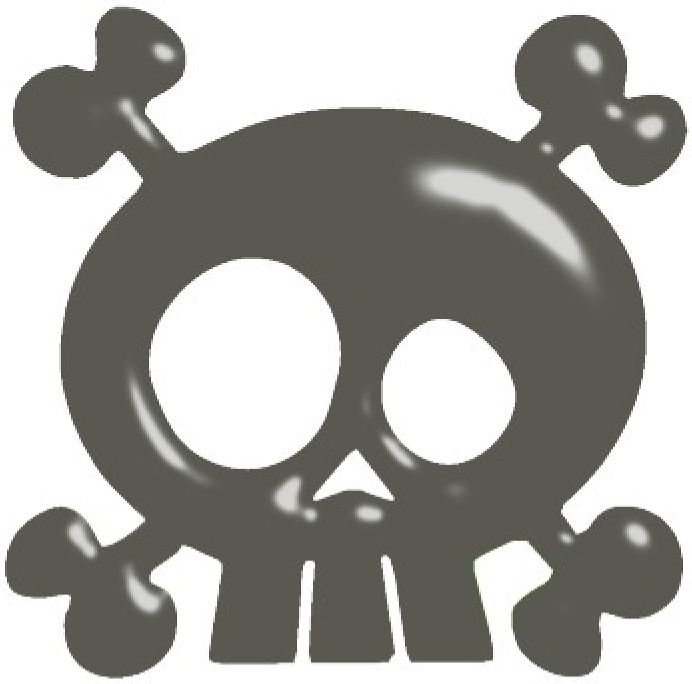
Normal map **N** **with regularization** (zoom)

Generation of the Illumination Layer

The illumination layer **J** is generated by applying the **Phong Illumination Model** (B.T. Phong (1975), “*Illumination for computer generated pictures*”):



Ambient light (**Ka**) +
Diffuse component (**Kd**)



Specular component (**Ks**)

Output Illumination Layer **J**

APPLICATION RESULTS

GREYC
Electronics and Computer Science Laboratory

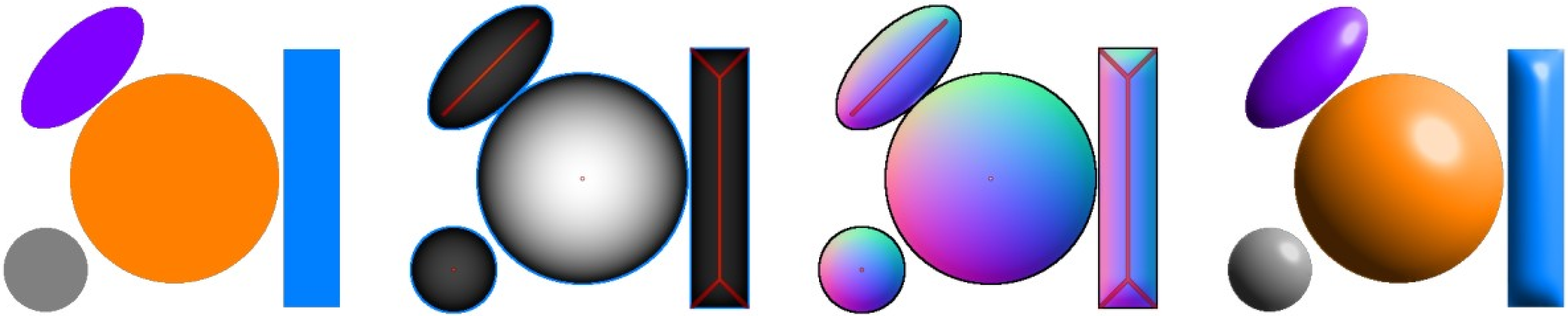
Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Result on Simple Shapes

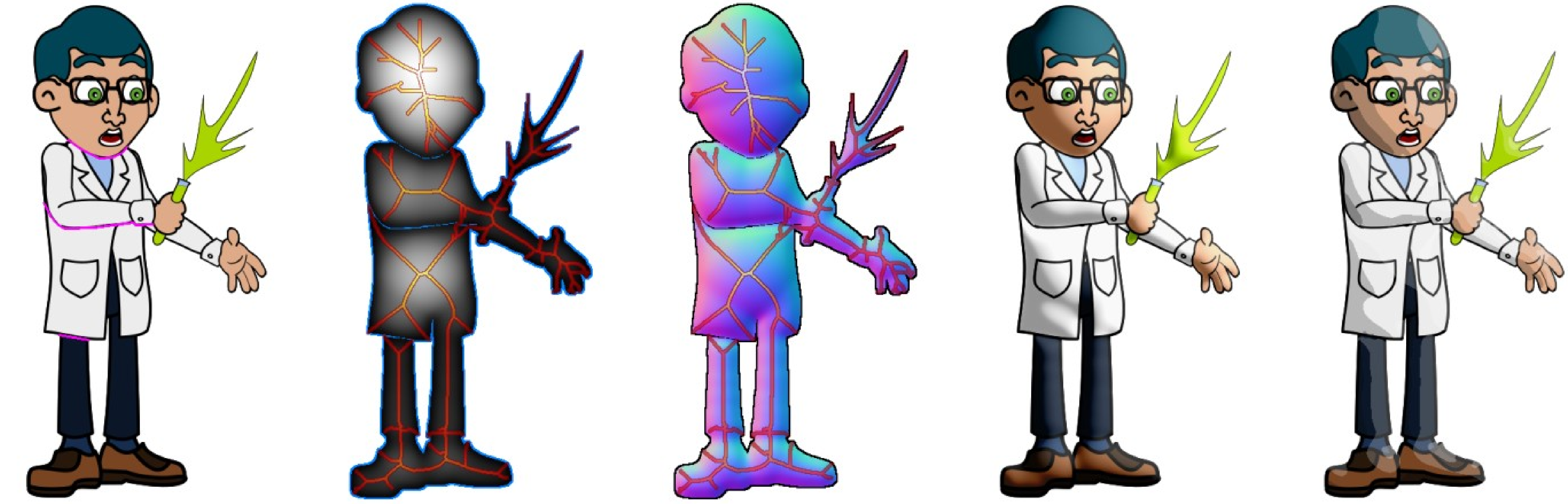


(a) Synthetic flat-colored drawing (b) Our estimated bumpmap (c) Our estimated normalmap (d) Our illumination result

Remark: A 2D silhouette with the shape of a **disk** will be illuminated as it is a **3D sphere**.

→ In that case, the skeleton **S** is reduced to a single point.
→ Only a single semi-sphere elevation is drawn in the bump map **B**.

Results on Real Images



(a) Flat-colored drawing with guiding lines (in magenta) *(b)* Our estimated bumpmap *(c)* Our estimated normalmap *(d)* Our illumination result *(e)* Our quantized illumination result (*cartoon* style)

Note: Different types of rendering for the illumination layer **J** can be obtained, by playing with the Phong model parameters and/or doing post-processing on **J** (e.g., *level quantization*, for result *(e)*).

Adding Guiding Lines

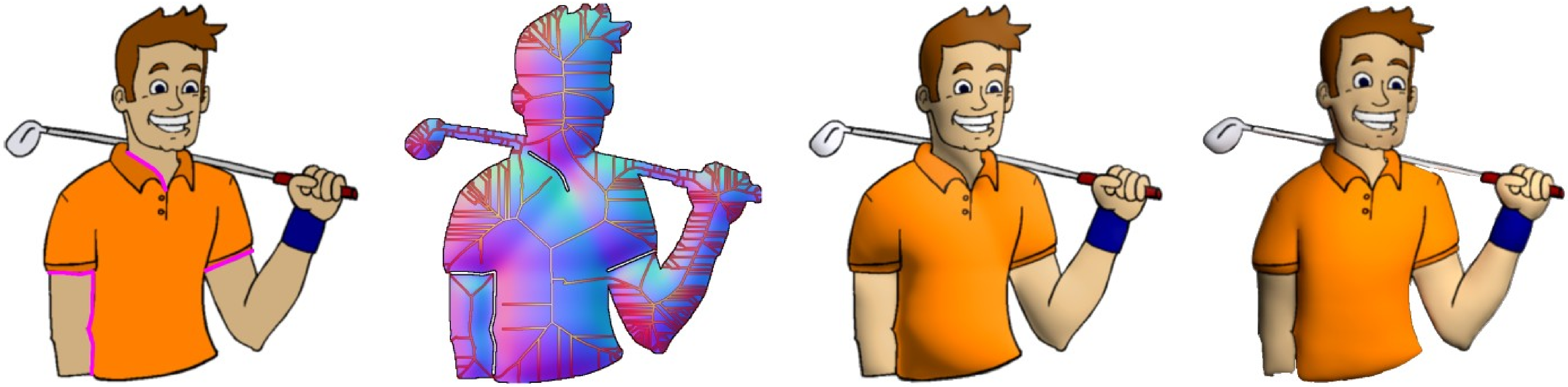
(*a*) Flat-colored drawing with guiding lines (in magenta) (*b*) Normalmap estimated from the silhouette only (*c*) Illumination generated from (*b*) (*d*) Normalmap taking into account guiding lines (*e*) Illumination generated from (*d*)

In the case of complex shapes with details inside, the user can add **guiding lines** (here shown in magenta), that are drawn as exterior pixels in the silhouette **S** before estimating the bump map **B**.

→ These pixels have 0-elevation in the estimated bump map **B**.
→ Explicitly add relief in the generated illumination layer **J.**

GREYC
Electronics and Computer Science Laboratory

Comparison With Neural-Net Approach



(*a*) Flat-colored drawing with guiding lines (in magenta)

(*b*) Our estimated bumpmap

(*c*) Our illumination result

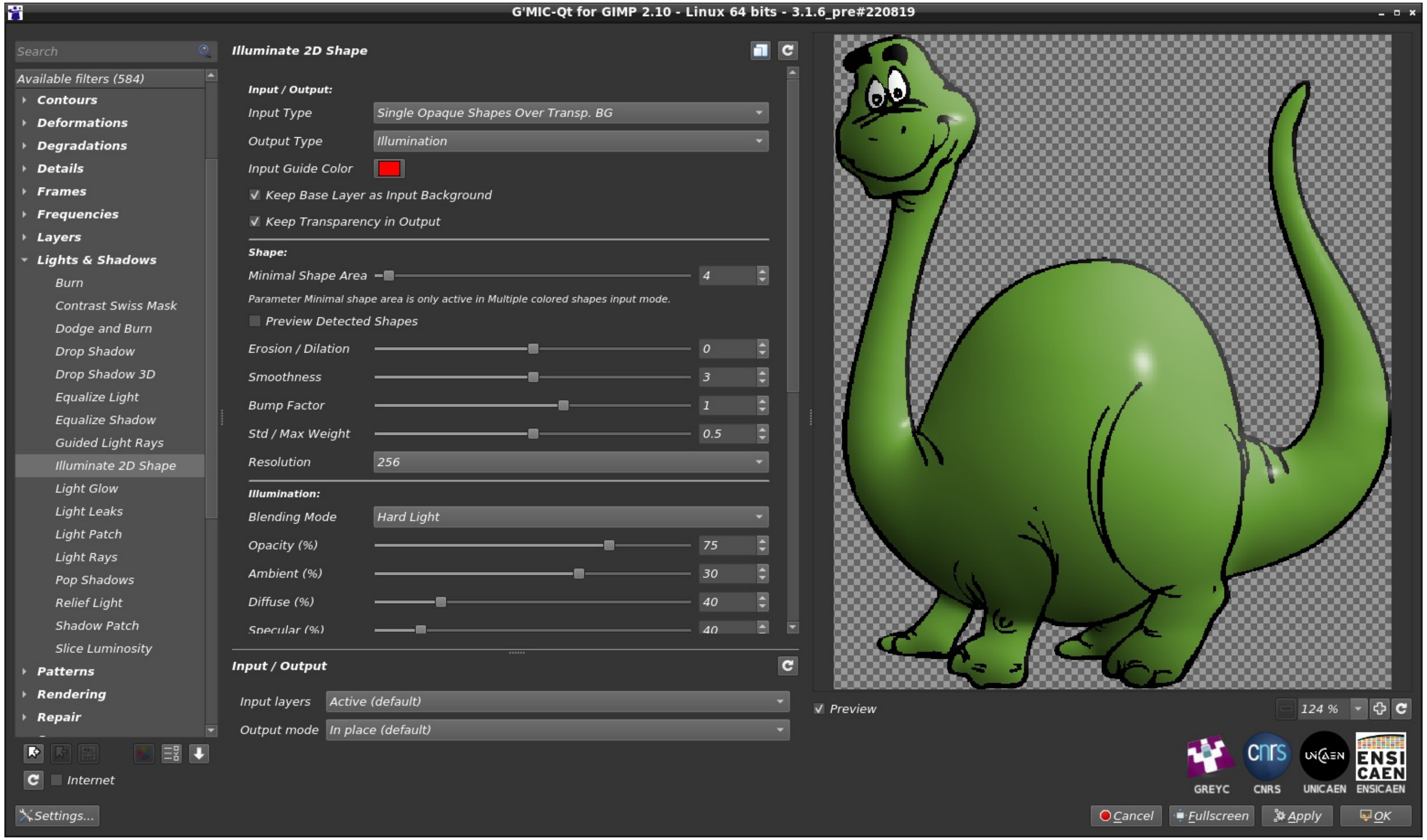
(*d*) Comparison: Result from neural network-based method

We compared our proposed method with:

M. Hudon, S. Lutz, R. Pagés, and A. Smolic (2019), “*Augmenting hand-drawn art with global illumination effects through surface inflation*,” in European Conference on Visual Media Production.

Results are very similar. However, our algorithm requires **no training** phase, **nor a lot of parameter** storage. It can run on a **low-cost machine**, without requirements for powerful GPUs.

Algorithm Integration Into G'MIC



Our algorithm has been integrated into **G'MIC**, an open-source framework for image processing.
You can test it on your own images (plug-in available for GIMP, Krita, Photoshop, Affinity Photo, …)

https://gmic.eu

Thanks for Your Attention !

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs