Learning to separate shading from paint

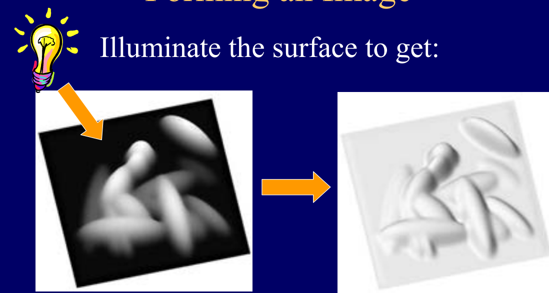
Marshall F. Tappen[1]
William T. Freeman[1]
Edward H. Adelson[1,2]

[1]MIT Computer Science and Ar
Intelligence Laboratory (CSA
[2]MIT Dept. of Brain and Cognitive Sciences

Marshall

Forming an Image

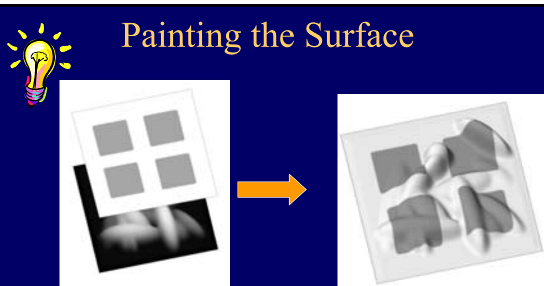
Illuminate the surface to get:



Surface | Shading Image

The "shading image" is the interaction of the shape of the surface and the illumination

Painting the Surface



Scene | Image

We can also include a reflectance pattern or a "paint" image. Now shading and reflectance effects combine to create the observed image.

Problem

How can we access shape or reflectance information from the observed image?

For example:

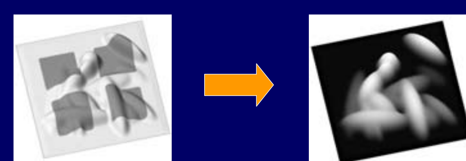


image | estimate of shape

Goal: decompose the image into shading and reflectance components.

Image = Shading Image × Reflectance Image

- These types of images are known as intrinsic images (Barrow and Tenenbaum).
- Note: while the images multiply, we work in a gamma-corrected domain and assume the images add.

Why you might want to compute these intrinsic images

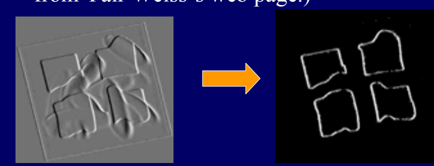
- Ability to reason about shading and reflectance independently is necessary for most image understanding tasks.
  - Material recognition
  - Image segmentation
- Want to understand how humans might do the task.
- An engineering application: for image editing, want access and modify the intrinsic images separately
- Intrinsic images are a convenient representation.
  - More informative than just the image
  - Less complex than fully reconstructing the scene

Treat the separation as a labeling problem

- We want to identify what parts of the image were caused by shape changes and what parts were caused by paint changes.
- But how represent that? Can't label pixels of the image as "shading" or "paint".
- Solution: we'll label *gradients* in the image as being caused by shading or paint.
- Assume that image gradients have only one cause.

Recovering Intrinsic Images

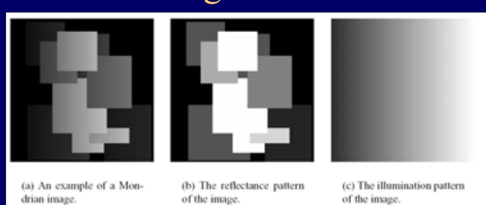
- Classify each *x* and *y* image derivative as being caused by *either* shading or a reflectance change
- Recover the intrinsic images by finding the least-squares reconstruction from each set of labeled derivatives. (Fast Matlab code for that available from Yair Weiss's web page.)



Original *x* derivative image

Classify each derivative (White is reflectance)

Classic algorithm: Retinex



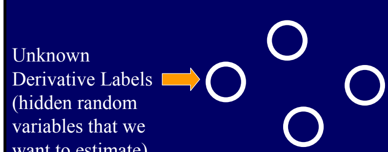
(a) An example of a Mondrian image. (b) The reflectance pattern of the image. (c) The illumination pattern of the image.

- Assume world is made up of Mondrian reflectance patterns and smooth illumination
- Can classify derivatives by the magnitude of the derivative

Outline of our algorithm (and the rest of the talk)

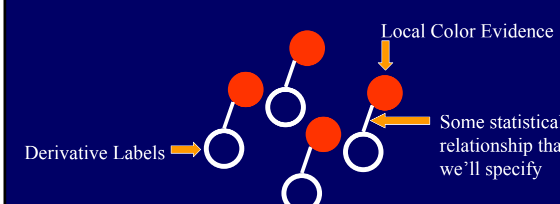
- Gather local evidence for shading or reflectance
  - Color (chromaticity changes)
  - Form (local image patterns)
- Integrate the local evidence across space.
  - Assume a probabilistic model and use belief propagation.
- Show results on example images

Probabilistic graphical model



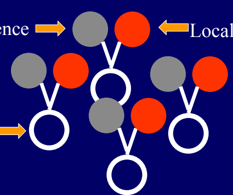
Probabilistic graphical model

- Local evidence



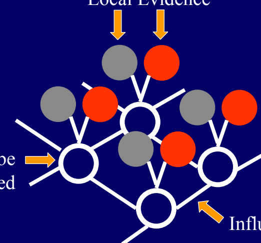
Probabilistic graphical model

- Local evidence



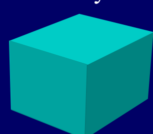
Probabilistic graphical model

Propagate the local evidence in Markov Random Field.
This strategy can be used to solve other low-level vision problems.



Local Color Evidence

For a Lambertian surface, and simple illumination conditions, shading only affects the intensity of the color of a surface



Notice that the chromaticity of each face is the same

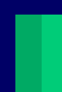
Any change in chromaticity must be a reflectance change

Classifying Color Changes

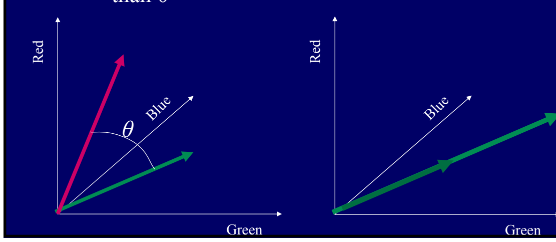
Chromaticity Changes

Angle between the two vectors, θ, is greater than 0

Intensity Changes

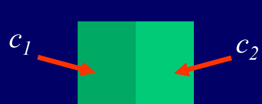


Angle between two vectors, θ, equals 0



Color Classification Algorithm

1. Normalize the two color vectors c_1 and c_2



2. If $(c_1 \cdot c_2) > T$
   - Derivative is a reflectance change
   - Otherwise, label derivative as shading

Result using only color information

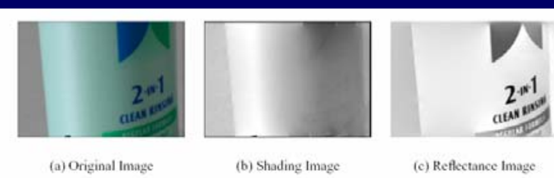


Figure 1: Example. Computed using Color Detector. To facilitate printing, the intrinsic images have been computed from a gray-scale version of the image. The color information is used solely for classifying derivatives in the gray-scale copy of the image.

Results Using Only Color

Input → Shading → Reflectance

- Some changes are ambiguous
- Intensity changes could be caused by shading or reflectance
  - So we label it as "ambiguous"
  - Need more information

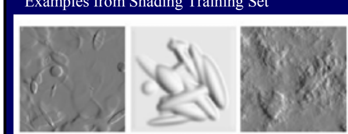
Utilizing local intensity patterns

- The painted eye and the ripples of the fabric have very different appearances
- Can learn classifiers which take advantage of these differences

Shading/paint training set

Examples from Reflectance Change Training Set

Examples from Shading Training Set



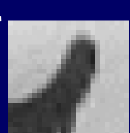
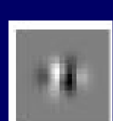
From Weak to Strong Classifiers: Boosting

- Individually these weak classifiers aren't very good.
- Can be combined into a single strong classifier.
- Call the classification from a weak classifier $h_i(x)$.
- Each $h_i(x)$ votes for the classification of x (-1 or 1).
- Those votes are weighted and combined to produce a final classification.

$$H(x) = \operatorname{sign}\left(\sum_i \alpha_i h_i(x)\right)$$

Using Local Intensity Patterns

- Create a set of weak classifiers that use a small image patch to classify each derivative
- The classification of a derivative:

abs ( I_p *  F) > T

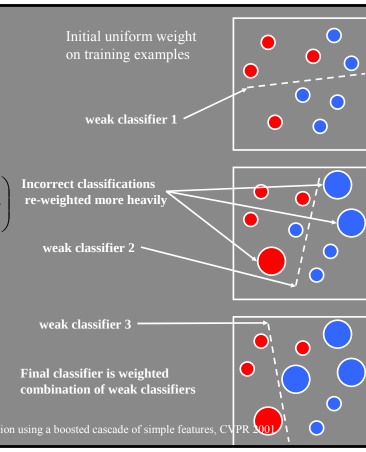
AdaBoost

(Freund & Shapire '95)

$$f(x) = \theta\left(\sum_t \alpha_t h_t(x)\right)$$

$$\alpha_t = 0.5 \log\left(\frac{error_t}{1 - error_t}\right)$$

$$w_t^i = \frac{w_{t-1}^i e^{-y_i \alpha_t h_t(x_i)}}{\sum_i w_{t-1}^i e^{-y_i \alpha_t h_t(x_i)}}$$



Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Use Newton's method to reduce classification cost over training set

Classification cost $J = \sum_i e^{-y_i f(x_i)}$

Treat h_m as a perturbation, and expand loss J to second order in h_m

$$\underbrace{\arg\min_{h_m} J}_{\text{cost function}}\underbrace{(H+h_m)}_{\text{classifier with perturbation}} \simeq \arg\min_{h_m} \sum_{c=1}^{C} E\Big[\underbrace{e^{-z^c H(v,c)}}_{\text{reweighting}}\underbrace{(z^c - h_m)^2}_{\text{squared error}}\Big]$$

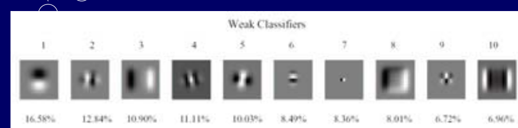
Adaboost demo…

Learning the Classifiers

- The weak classifiers, $h_i(x)$, and the weights α are chosen using the *AdaBoost* algorithm (see www.boosting.org for introduction).
- Train on synthetic images.
- Assume the light direction is from the right.

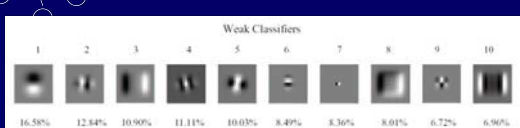
- Filters for the candidate weak classifiers—cascade two out of these 4 categories:
  - Multiple orientations of 1st derivative of Gaussian filters
  - Multiple orientations of 2nd derivative of Gaussian filters
  - Several widths of Gaussian filters
  - impulse

Classifiers Chosen



- These are the filters chosen for classifying vertical derivatives when the illumination comes from the top of the image.
- Each filter corresponds to one $h_i(x)$

Characterizing the learned classifiers



- Learned rules for all (but classifier 9) are: if rectified filter response is above a threshold, vote for reflectance.
- Yes, contrast and scale are all folded into that. We perform an overall contrast normalization on all images.
- Classifier 1 (the best performing single filter to apply) is an empirical justification for Retinex algorithm: treat small derivative values as shading.
- The other classifiers look for image structure oriented perpendicular to lighting direction as evidence for reflectance change.

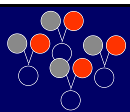
Results Using Only Form Information

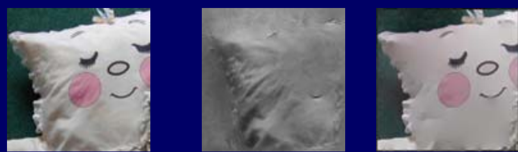

Input Image


Shading Image


Reflectance Image

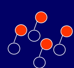
Using Both Color and Form Information

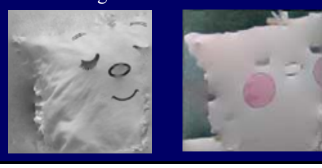




Input image | Shading | Reflectance

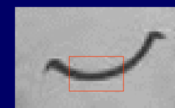
Results only using chromaticity.





Some Areas of the Image Are Locally Ambiguous

Input



Is the change here better explained as

Shading

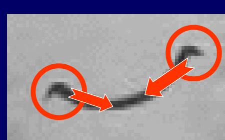
or

Reflectance

?

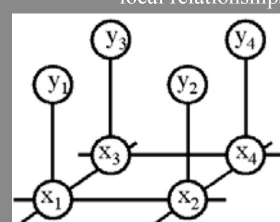
Propagating Information

- Can disambiguate areas by propagating information from reliable areas of the image into ambiguous areas of the image



Markov Random Fields

- Allows rich probabilistic models for images.
- But built in a local, modular way. Learn local relationships, get global effects out.



Network joint probability

$$P(x,y)=\frac{1}{Z}\prod_{i,j}\Psi(x_i,x_j)\prod_{i}\Phi(x_i,y_i)$$

scene

image

Scene-scene compatibility function

neighboring scene nodes

Image-scene compatibility function

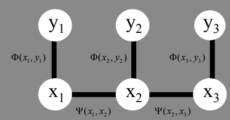
local observations

Inference in MRF's

- Inference in MRF's. (given observations, how infer the hidden states?)
  - Gibbs sampling, simulated annealing
  - Iterated condtional modes (ICM)
  - Variational methods
  - Belief propagation
  - Graph cuts

See www.ai.mit.edu/people/wtf/learningvision for a tutorial on learning and vision.

Derivation of belief propagation

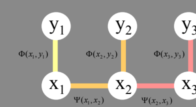


$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1,x_2,x_3,y_1,y_2,y_3)$$

The posterior factorizes

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1,x_2,x_3,y_1,y_2,y_3)$$

$$= \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1,y_1)\ \Phi(x_2,y_2)\ \Psi(x_1,x_2)\ \Phi(x_3,y_3)\ \Psi(x_2,x_3)$$

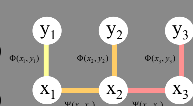


Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1,x_2,x_3,y_1,y_2,y_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1,y_1)\ \Phi(x_2,y_2)\ \Psi(x_1,x_2)\ \Phi(x_3,y_3)\ \Psi(x_2,x_3)$$

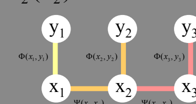
$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \Phi(x_1,y_1)\ \underset{x_2}{\text{sum}}\ \Phi(x_2,y_2)\ \Psi(x_1,x_2)\ \underset{x_3}{\text{sum}}\ \Phi(x_3,y_3)\ \Psi(x_2,x_3)$$



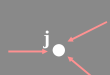
Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \Phi(x_1,y_1)\ \underset{x_2}{\text{sum}}\ \Phi(x_2,y_2)\ \Psi(x_1,x_2)\ \underset{x_3}{\text{sum}}\ \Phi(x_3,y_3)\ \Psi(x_2,x_3)$$

$$M_1^2(x_1) = \underset{x_2}{\text{sum}}\ \Psi(x_1,x_2)\ \Phi(x_2,y_2)\ M_2^3(x_2)$$

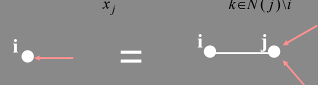


Belief, and message updates



$$b_j(x_j) = \prod_{k\in N(j)} M_j^k(x_j)$$

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i,x_j) \prod_{k\in N(j)\backslash i} M_j^k(x_j)$$

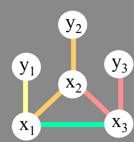


Optimal solution in a chain or tree: Belief Propagation

- “Do the right thing” Bayesian algorithm.
- For Gaussian random variables over time: Kalman filter.
- For hidden Markov models: forward/backward algorithm (and MAP variant is Viterbi).

No factorization with loops!

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\, \Phi(x_1, y_1)\ \underset{x_2}{\text{sum}}\, \Phi(x_2, y_2)\, \Psi(x_1, x_2)\ \underset{x_3}{\text{sum}}\, \Phi(x_3, y_3)\, \Psi(x_2, x_3)\ \Psi(x_1, x_3)$$

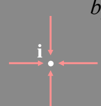


Justification for running belief propagation in networks with loops

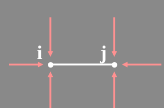
- Experimental results:
  - Error-correcting codes — Kschischang and Frey, 1998; McEliece et al., 1998
  - Vision applications — Freeman and Pasztor, 1999; Frey, 2000
- Theoretical results:
  - For Gaussian processes, means are correct. — Weiss and Freeman, 1999
  - Large neighborhood local maximum for MAP. — Weiss and Freeman, 2000
  - Equivalent to Bethe approx. in statistical physics. — Yedidia, Freeman, and Weiss, 2000
  - Tree-weighted reparameterization — Wainwright, Willsky, Jaakkola, 2001

Region marginal probabilities

$$b_i(x_i) = k\, \Phi(x_i) \prod_{k \in N(i)} M_i^k(x_i)$$

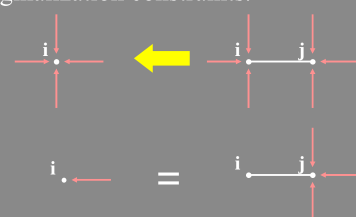


$$b_{ij}(x_i, x_j) = k\, \Psi(x_i, x_j) \prod_{k \in N(i)\backslash j} M_i^k(x_i) \prod_{k \in N(j)\backslash i} M_j^k(x_j)$$



Belief propagation equations

Belief propagation equations come from the marginalization constraints.



$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j)\backslash i} M_j^k(x_j)$$

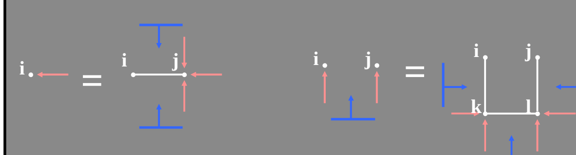
Results from Bethe free energy analysis

- Fixed point of belief propagation equations iff. Bethe approximation stationary point.
- Belief propagation always has a fixed point.
- Connection with variational methods for inference: both minimize approximations to Free Energy,
  - **variational**: usually use primal variables.
  - **belief propagation**: fixed pt. equs. for dual variables.
- Kikuchi approximations lead to more accurate belief propagation algorithms.
- Other Bethe free energy minimization algorithms—Yuille, Welling, etc.

Kikuchi message-update rules

Groups of nodes send messages to other groups of nodes.

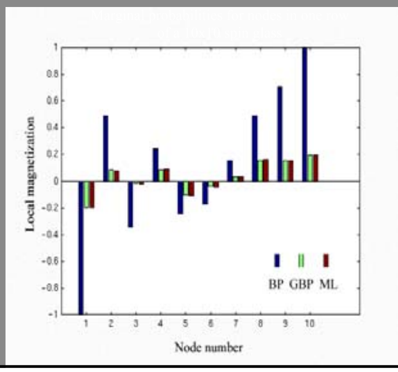
Typical choice for Kikuchi cluster.



Update for messages

Update for messages

Generalized belief propagation

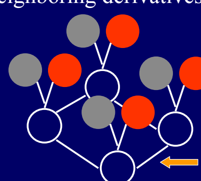


References on BP and GBP

- J. Pearl, 1985
  - classic
- Y. Weiss, NIPS 1998
  - Inspires application of BP to vision
- W. Freeman et al learning low-level vision, IJCV 1999
  - Applications in super-resolution, motion, shading/paint discrimination
- H. Shum et al, ECCV 2002
  - Application to stereo
- M. Wainwright, T. Jaakkola, A. Willsky
  - Reparameterization version
- J. Yedidia, AAAI 2000
  - The clearest place to read about BP and GBP.

Propagating Information

- Extend probability model to consider relationship between neighboring derivatives



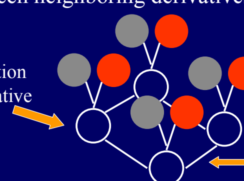
$$\psi(x_i, x_j) = \begin{bmatrix} \beta & 1-\beta \\ 1-\beta & \beta \end{bmatrix}$$

- β controls how necessary it is for two nodes to have the same label
- Use Generalized Belief Propagation to infer labels. (Yedidia et al. 2000)

Propagating Information

- Extend probability model to consider relationship between neighboring derivatives

Classification of a derivative



$$\psi(x_i, x_j) = \begin{bmatrix} \beta & 1-\beta \\ 1-\beta & \beta \end{bmatrix}$$

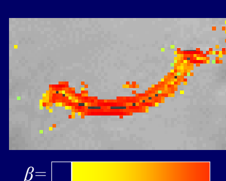
- β controls how necessary it is for two nodes to have the same label
- Use Generalized Belief Propagation to infer labels. (Yedidia et al. 2000)

Setting Compatibilities

- All compatibilities have form

$$\psi(x_i, x_j) = \begin{bmatrix} \beta & 1-\beta \\ 1-\beta & \beta \end{bmatrix}$$

- Assume derivatives along image contours should have the same label
- Set β close to 1 when the derivatives are along a contour
- Set β to 0.5 if no contour is present
- β is computed from a linear function of the image gradient's magnitude and orientation

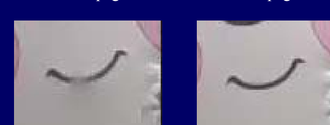


Improvements Using Propagation

Input Image

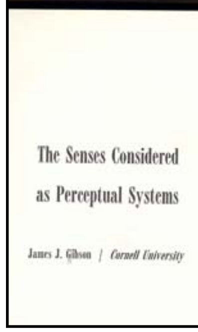
Reflectance Image Without Propagation

Reflectance Image With Propagation



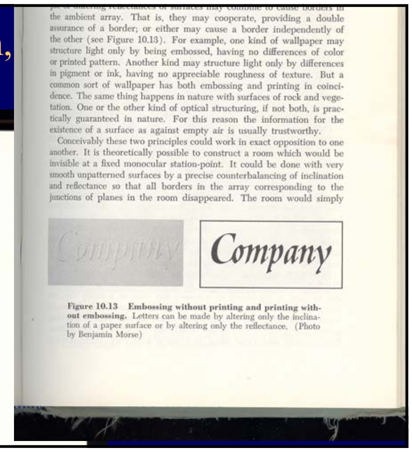
More results…

J. J. Gibson, 1968



The Senses Considered as Perceptual Systems

James J. Gibson / Cornell University



the ambient array. That is, they may cooperate, providing a double assurance of a border; or either may cause a border independently of the other (see Figure 10.13). For example, one kind of wallpaper may structure light only by being embossed, having no differences of color or printed pattern. Another kind may structure light only by differences in pigment or ink, having no appreciable roughness of texture. But a common sort of wallpaper has both embossing and printing in coincidence. The same thing happens in nature with surfaces of rock and vegetation. One or the other kind of optical structuring, if not both, is practically guaranteed in nature. For this reason the information for the existence of a surface as against empty air is usually trustworthy.

Conceivably these two principles could work in exact opposition to one another. It is theoretically possible to construct a room which would be invisible at a fixed monocular station-point. It could be done with very smooth unpatterned surfaces by a precise counterbalancing of inclination and reflectance so that all borders in the array corresponding to the junctions of planes in the room disappeared. The room would simply

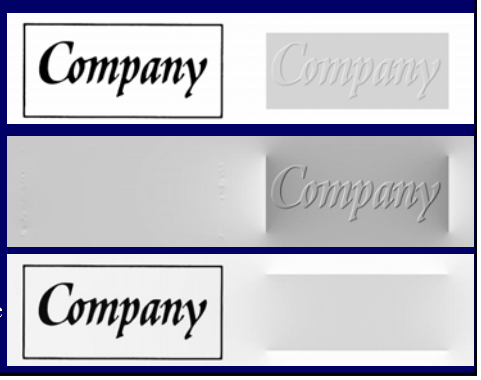
Figure 10.13 Embossing without printing and printing without embossing. Letters can be made by altering only the inclination of a paper surface or by altering only the reflectance. (Photo by Benjamin Morse)

Gibson image

original

shading

reflectance



Clothing catalog image

Original
(from LL Bean catalog)

Shading

Reflectance

Sign at train crossing

Separated images

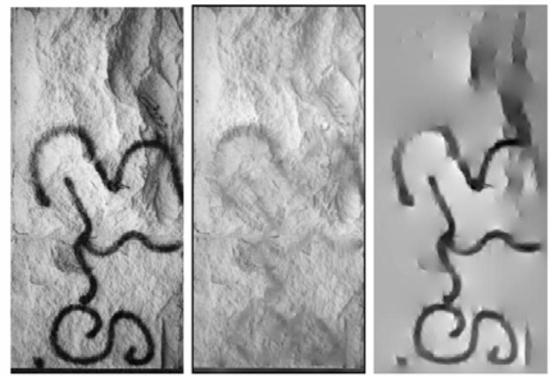
original

shading

reflectance

Note: color cue omitted for this processing

(a) Original Image



(a) Original Image (b) Shape Image (c) Reflectance Image

Finally, returning to our explanatory example…

input

Ideal shading image

Ideal paint image

Algorithm output.
Note: occluding edges labeled as reflectance.

Summary

- Sought an algorithm to separate shading and reflectance image components.
- Achieved good results on real images.
- Classify local derivatives
  - Learn classifiers for derivatives based on local evidence, both color and form.
- Propagate local evidence to improve classifications.

For manuscripts, see www.ai.mit.edu/people/wtf/