Recovering Intrinsic Images from a Single Image

by Marshall F. Tappen, William T. Freeman, Edward H. Adelson

Andriy Mnih
CSC 2541
February 11, 2005

Intrinsic images

- Segmentation and object recognition algorithms often have trouble dealing the various illumination effects in images.
- Removing the “shading component” from input images can significantly improve the performance of such algorithms.
- Computing an intrinsic image representation of a given image, which consists of a reflectance image and a shading image, is one way to achieve this.

Some definitions

- Reflectance is the ratio of incoming light to the reflected light at a point. Informally, reflectance can be thought of as the colour of the point on a surface when illuminated uniformly with white light from all directions.
- Shading is what happens when light interacts with surfaces.
- An image is modelled as a product of its reflectance image and its shading image: $I(x, y) = R(x, y) S(x, y)$
- From this formula it is clear that reflectance image can only be recovered up to a multiplicative factor unless some additional information is available.

Outline of the method

1. Transform the image into the log domain.
2. Apply derivative filters to the transformed image.
3. Split each of the resulting derivative images into a shading derivative image and a reflectance derivative image by classifying each pixel (derivative) as caused by shading or a change in reflectance.
4. Reconstruct the shading image and the illumination image from their derivative images.

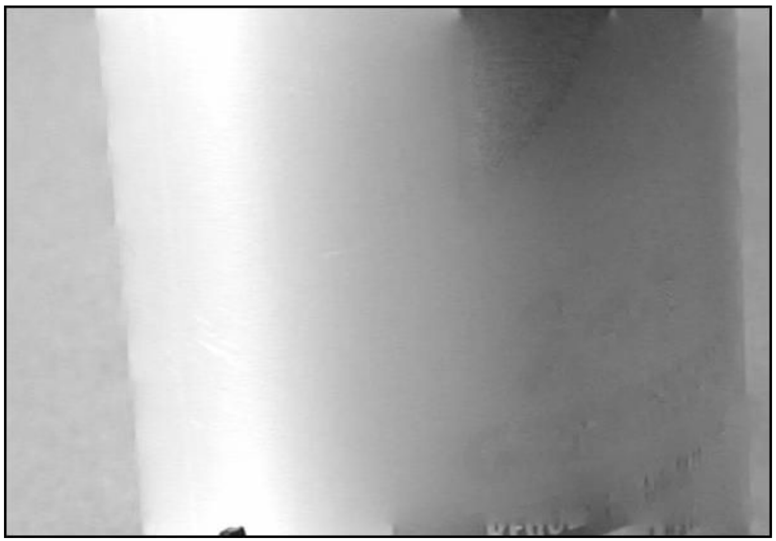
Assumptions

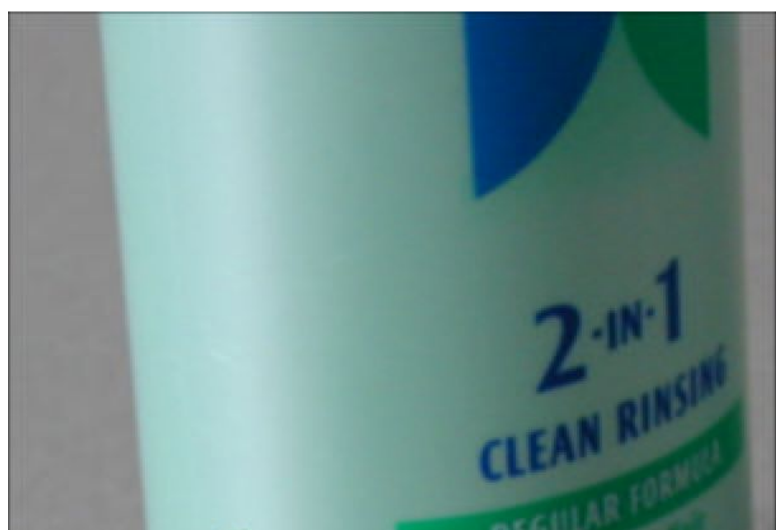
1. Each image derivative (*x* or *y*) is caused either by shading or a change in reflectance, but not both.
2. It is possible to infer these causes of image derivatives reliably.

Using colour information

- If all surfaces in the scene are diffuse, a change in colour (as opposed to a change in colour intensity) between neighbouring pixels indicates a change in reflectance. This is true only if all lights are white.
- To classify a derivative, a dot product of the normalized colour RGB vectors of two nearby pixels is computed. If this dot product is below a threshold, the derivative is classified as a reflectance derivative.
- Which two pixels are used to classify the derivative at the given location is not clear.
- It is also not clear whether thresholds are learned or tuned by hand.

Results: using only colour information





Using grey-scale information

- A change in colour intensity can be caused either by shading or by a change in reflectance.
- The colour-based classifier cannot classify such changes correctly.
- Idea: Classify each derivative based on a grey-scale version of the image patch centered at the derivative location.
- This approach assumes that (grey-scale) shading patterns can be discriminated reliably from reflectance patterns.

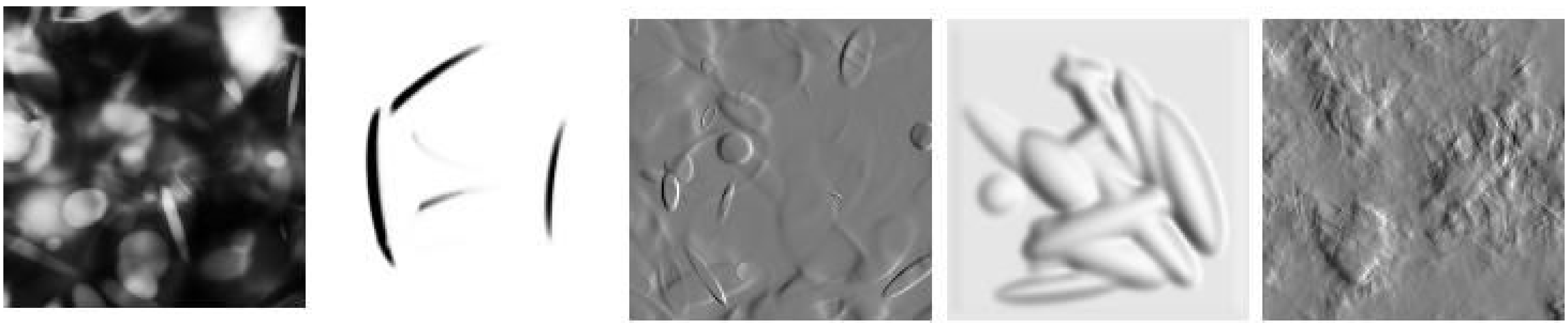
Grey-scale classifier

- The classifier is built by combining simple classifiers using the AdaBoost algorithm.
- Each simple classifier compares the output of a non-linear filter to a threshold, where the filter output is given by $F = \left| I_p * w \right|$.
- A simple classifier is "trained" by selecting a filter from a set of oriented derivative of Gaussian filters, which gives the lowest error on a (weighted) training set.
- It is not clear how the threshold is learned.

Training the classifier

- AdaBoost trains simple classifiers, one at a time, on weighted training sets. The training cases that were misclassified by the previously trained classifiers are given larger weights.
- The classifier output is obtained by weighting the outputs of the simple classifiers by a measure of their performance.
- It seems that the same grey-scale classifier is used for both x and y derivatives.

Training data

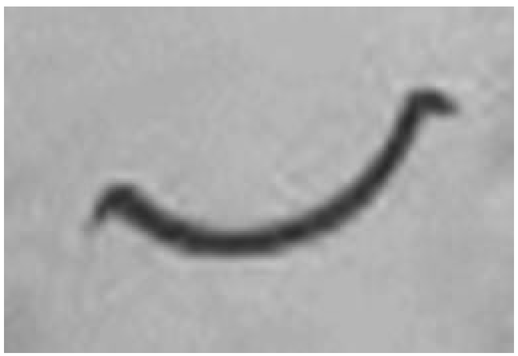


- The training set contained images of fractal surfaces, and randomly placed ellipses and lines. Each image was either a reflectance or a shading image.
- Illumination in all images in the training set was from the right.

Results: using only grey-scale information

Why information propagation is necessary



- Both the colour and the grey-scale classifiers look only at small image patches.
- Such local information can be ambiguous.
- Propagating information between regions can be used to resolve the ambiguity.

Propagating evidence

- Idea: Treat each derivative label (shading/reflectance) as a node in a Markov Random Field. Set up the compatibility functions between the nodes to prefer neighbouring nodes lying along an image contour to have identical labels.
- Use a different MRF for each derivative filter (x or y).
- Each node is connected to four other nodes, forming a grid.
- The compatibility functions depend on the image.

Compatibility functions

- The compatibility function is defined as

$$\psi(x_i, x_j) = \begin{cases} \beta & \mathit{if}\ x_i = x_j \\ 1-\beta & \mathit{if}\ x_i \neq x_j \end{cases}$$

$$\beta(I_{xy}) = \frac{1}{1+\exp(-z(I_{xy}))}$$

where z is a function of the magnitude of the image gradient and the angle the gradient makes with the graph edge for which the compatibility function is defined.

Learning the compatibility functions

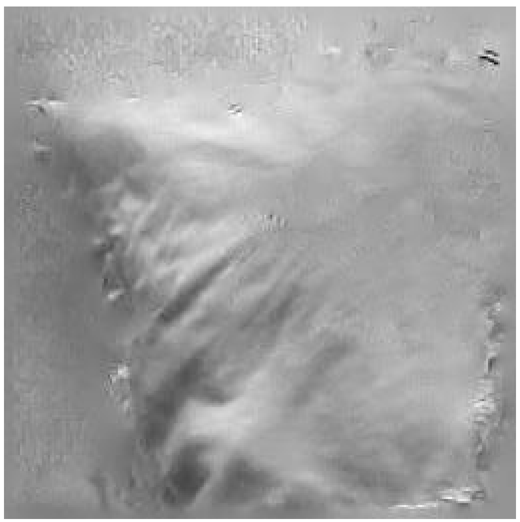
- Compatibility function parameters are learned by maximizing the probability of the training set
$$P = \frac{1}{Z} \prod_{(i,j)} \psi(x_i, x_j),$$
while pretending that the normalizing term Z is a constant, which it is not.
- The resulting z is $z = -1.2 \times \phi + 1.62 \times |\triangle I| + 2.3,$
where ϕ and $|\triangle I|$ are normalized to be between 0 and 1.
- When the gradient is smaller than 0.05, β is set to 0.5.

Inferring label values

- The local evidence at each node is computed by combining predictions of the colour and grey-scale classifiers.
  - Independence of the predictions is assumed.
- Node labels in the MRF are inferred using the Generalized Belief Propagation algorithm.
  - GBP is like BP, but the messages are sent between groups of nodes, instead of individual nodes.
  - GBP performance depends heavily on the choice of the node groups.
  - The paper does not say anything about the groups used.

Results (I)

Original image



Shading image

Reflectance image

Results (II)

Observations

- Working only with images lit from the right probably makes the task of classifying derivatives much easier than it would be in general.
- Strangely, at least one test image was lit from the left.
- The algorithm was unable to classify large shadows properly.
- Parts of reflectance images are sometimes very blurry.
- It is not clear how well the method handles textured surfaces.
- The method will almost certainly fail on images with sharp reflections.

Questions

- Is there a way to incorporate natural image statistics-based priors into this method?
- Would a more powerful classifier improve performance significantly?
- How realistic is the assumption that shading patterns can be distinguished from reflectance patterns reliably?
- Is the physics-free approach to computing intrinsic images the right way to go?