Ali Bilgin Arslan

Using 3D Cues in Face Perception

Project Abstract

Extracting 3D topography of objects using RGB images is a topic that is increasingly becoming popular in vision research. A series of techniques have been developed to recover 3d shape from surface properties, one of which makes use of sheen on the completely reflective surfaces (Adelson, E H, Torralba, A, Fleming, R W, 2009). In this project, I will be using a hierarchical model of the ventral stream, the HMAX model T. (Serre, 2007) to investigate the feasibility of deriving the 3d curvature of a natural object category, human faces, simply by using the orientation selective units that are analogous to the V1 cortical columns in primates.
Adelson et al (2009) showed that the surface curvature can be recovered by using the deformations on the surface of the object (figure 1)

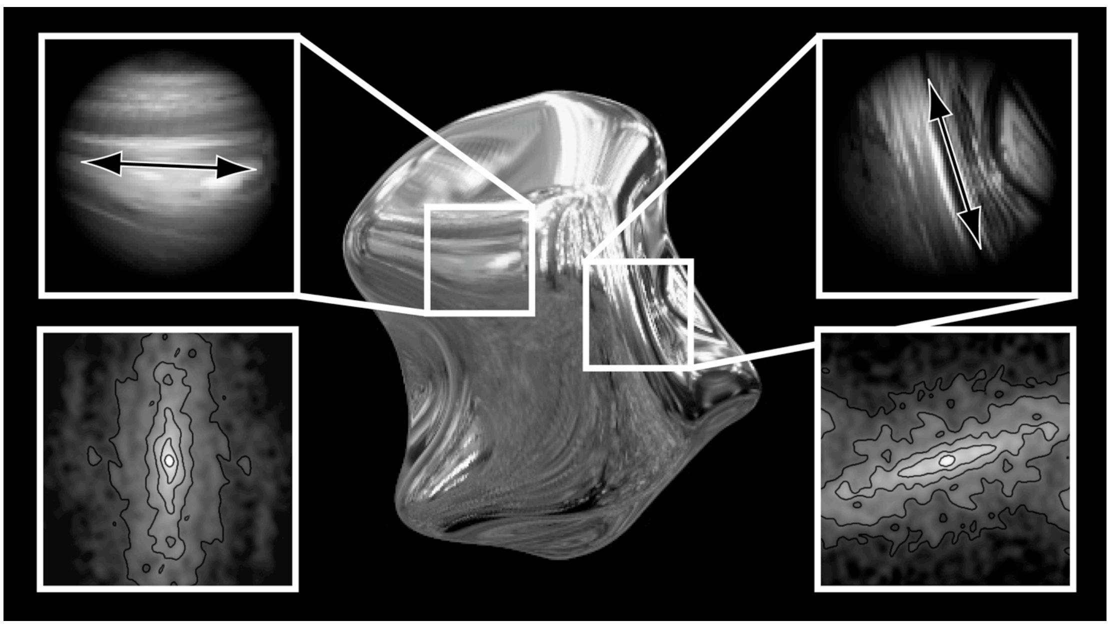

Figure 1: A completely reflective object in a textured environment. The Fourier spectrum of the small reflective patches cropped on the object reveal the curvature of the underlying surface. Taken from Adelson et al (2009).

Even though recovering the curvature of a completely reflective surface can be possible under limited conditions (i.e. when the object is placed in an environment that casts a uniform reflection on the object and where the lighting is not occluding parts of the object), it has been shown that it is possible to recover information about the 3d geometry of the object in other conditions as well, such as when the object is covered with a uniform procedural texture or when it is completely non-reflective, as long as the surface anisotropy can be measured without any significant confounds. (figure 2)

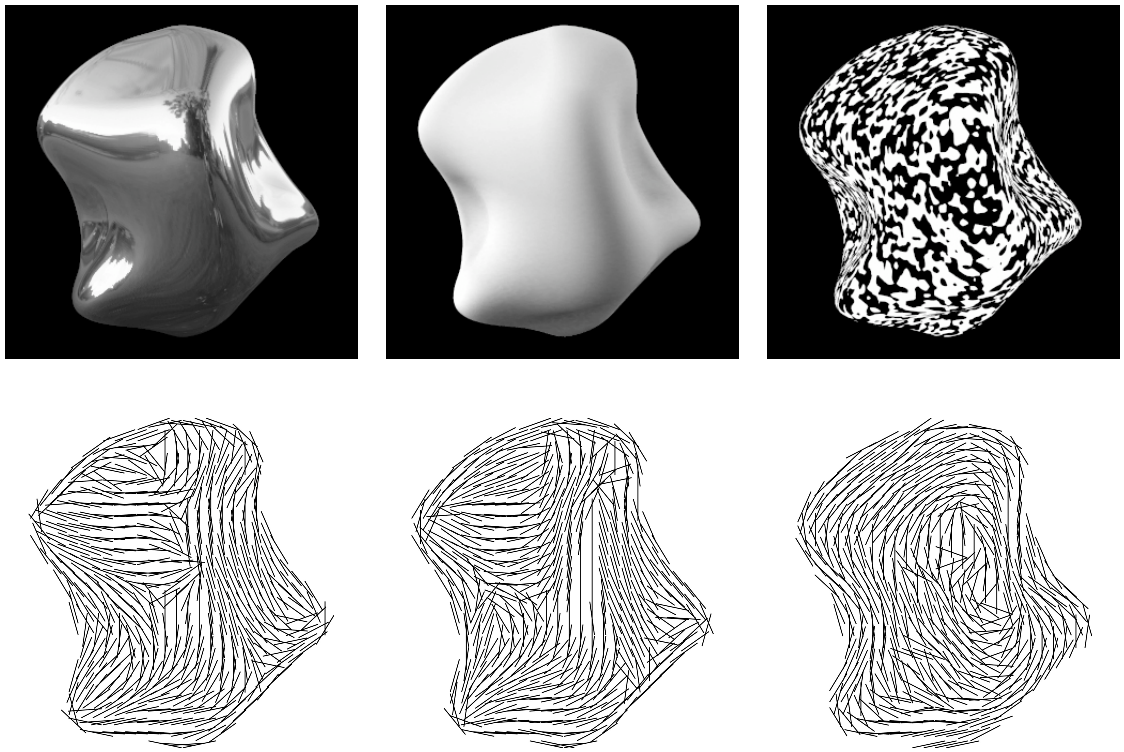

Figure 2: 3 different renderings of the same object, containing information about the geometry of the underlying object. Taken from Adelson et al (2009).

Since the present study focuses on the faces, which can only be represented in a high dimensional space due to the large number of variables, it is critically important to establish a database with reliable ground truths. For this purpose, instead of using a face database that is readily available online, I decided to use a database that I designed myself. The reason why the faces are chosen for this task is that despite the high variability of each individual face, they all share an underlying geometrical template with a sizable amount of variation.

This database includes realistic renderings of human faces created parametrically in a 3D modeling environment (FaceGen). Each "individual" in the database have these renderings:

- Original renderings: RGB renderings containing the actual picture of the individuals, in different lighting conditions, different viewing and camera angles.
- Surface normals: RGB images showing the orientation of each polygon in a 3D space. The orientations are coded in the color channels (i.e. blue channel corresponding to the orientation in Z axis.) and are relative to the camera's viewing angles.

- Z-Depth maps: A Grayscale image showing the distance of each point of the 3D model to the camera.

Each original rendering has a corresponding surface normal and depth map, which essentially enables the user to express each single pixel on the original renderings not only in RGB values, but also as a point in a 3D space (figure 3).

Figure 3: Sample face from the database: original render(left), surface normals showing the orientation (middle), depth map (left)

The processing steps are as follows:

Producing data for regression:

1. Compute S1 maps of the original render (from many individuals in different viewing and lighting angles) using the MATLAB implementation of HMAX. This will filter the image using a bank of Gaussian filters in 12 orientations, and 8 different sizes.
2. Crop patches from the S1 maps from randomly chosen locations on the face.
3. Vectorize these patches so that each vector contains 12*8*patchSize^2 elements.
4. Crop patches from the surface normals corresponding each original render used to create S1 maps, from the exact same locations.
5. Derive the XYZ orientation from each surface normal patch and average the values so that each patch is represented with a 3-element orientation vector.

Regression procedure

1. Divide the data into training (75%) and test sets (25%) that are exclusive.
2. Using Support Vector Regression (included in the libsvm package), train a regression model. Since the "label" for each "observation"(S1 response for a single patch) is a 3-element vector, this is essentially a vector regression.
3. Use the test set, assess the validity of the regression model.
4. Repeat the steps above to establish a confidence interval for the accuracy.

Predictions

The original renders incorporate a large amount of variation in terms of skin color, texture, and surface topography. Training a regressor using this database can provide a model with is invariant to the color and texture of the image and sensitive to the orientation of the surface.

The hypothesis is that the S1 units, which mimic the activity of orientation selective cells in V1 will provide cues about the curvature of the underlying surface simply by utilizing geometrical deformations in the texture, much like inferring the surface geometry from sheen.

In a neurolobiological perspective, positive result of this project would be insufficient to make any conclusive comments about the 3d perception in the primary visual cortex. It is know that the visual system utilizes many other information modalities such as stereo cues, depth and statistical properties of the natural images which may or may not be available at the level of V1. On the other hand, if simple orientation selectivity does provide cues about 3D shape, it is possible to postulate that the visual system starts building up a 3D representation of a scene as early as the level of V1.

References

- Adelson, E H, Torralba, A; Fleming, R W (2009). Shape from Sheen. MIT-CSAIL-TR-2009-051, Technical Report.
- T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio. Object recognition with cortex-like mechanisms. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (3), pp. 411-426 , 2007