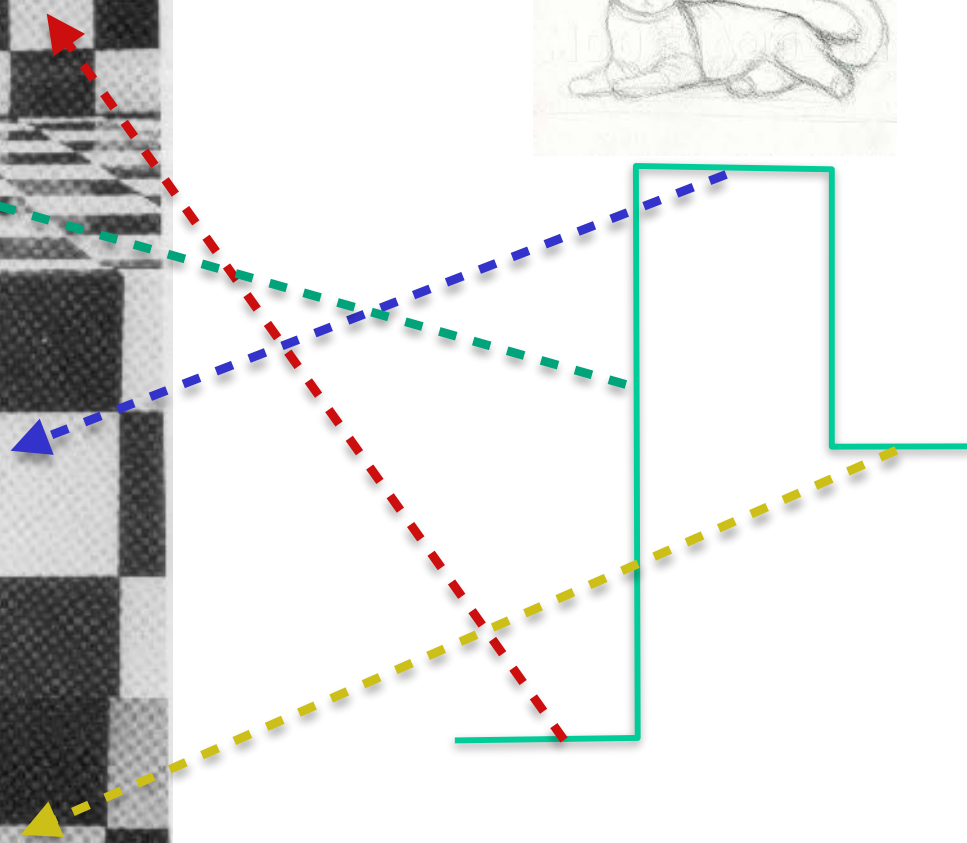
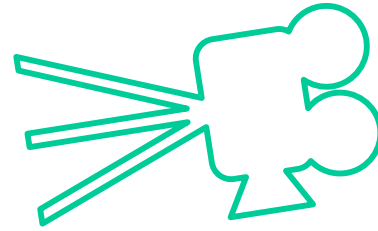
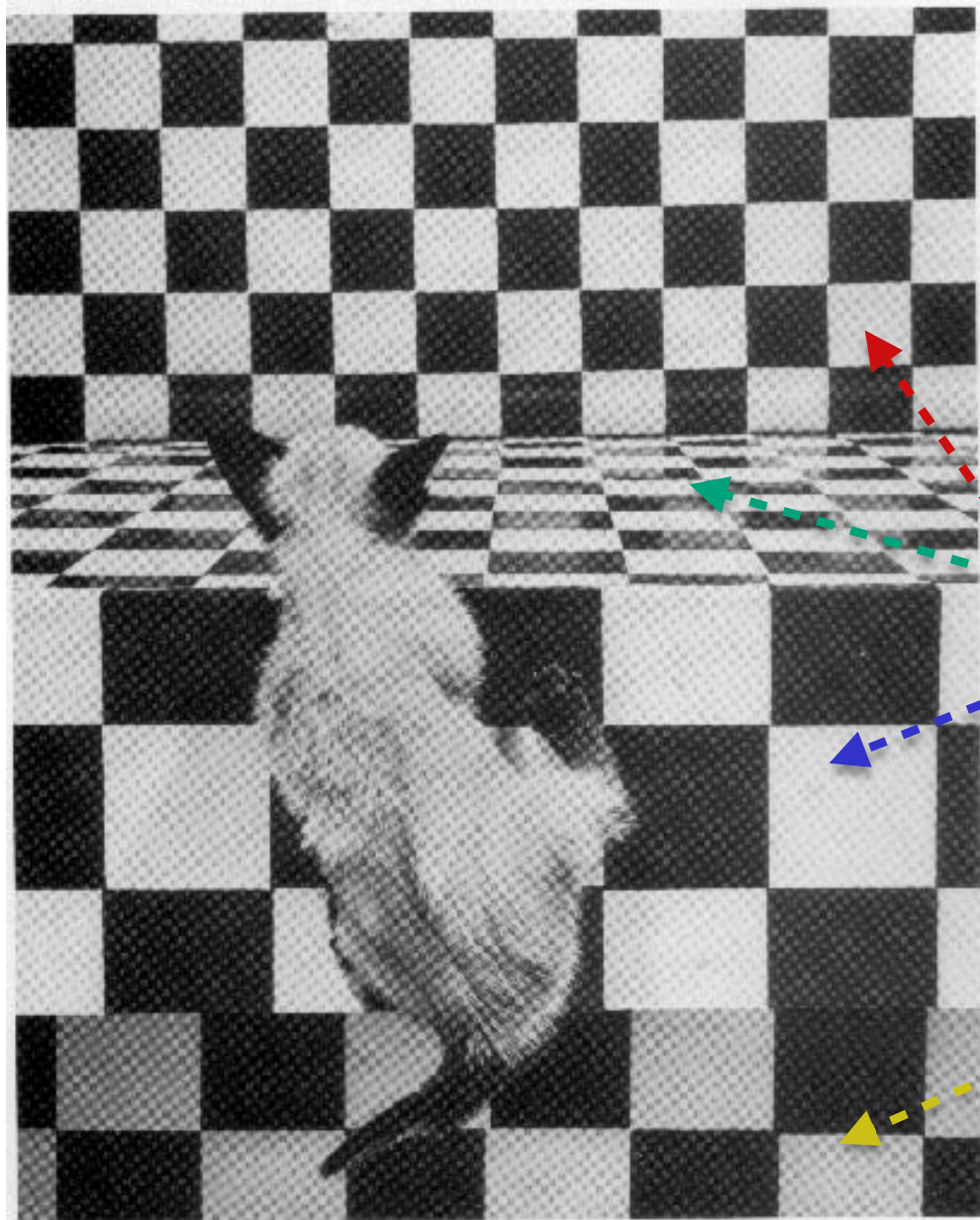


# Shape From Texture



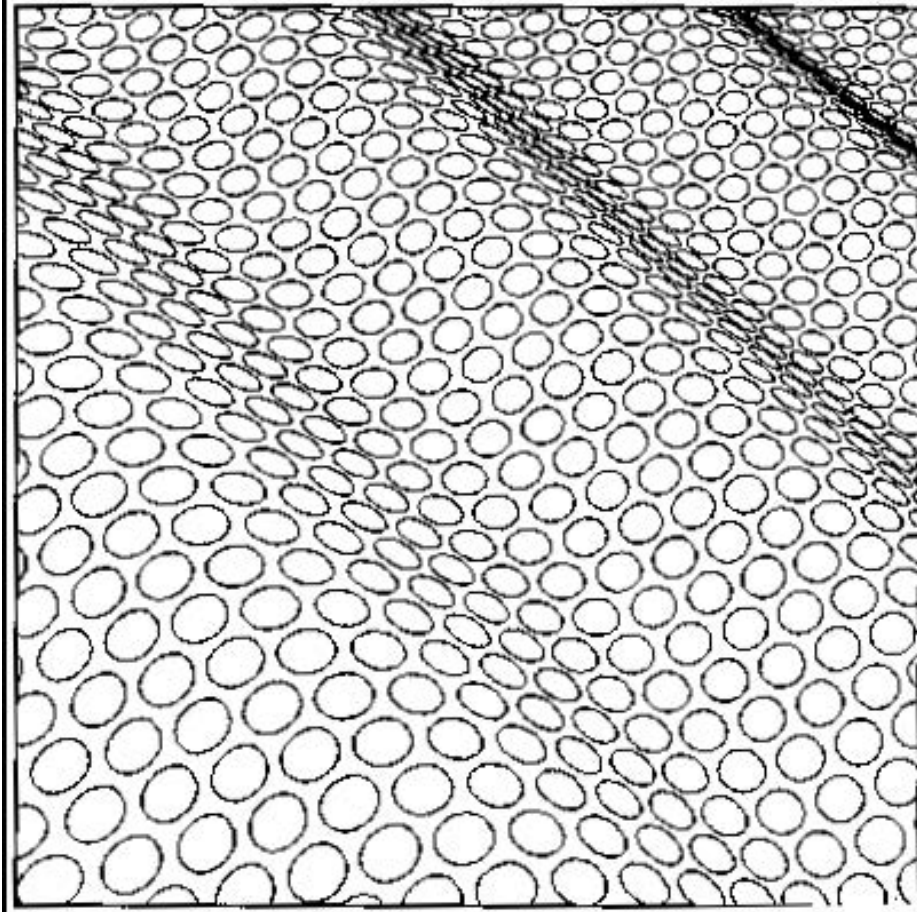
# Shape From Texture



Recover surface orientation or surface shape from image texture:

- Assume texture 'looks the same' at different points on the surface.
- This means that the deformation of the texture is due to the surface curvature.

# Structural Shape Recovery

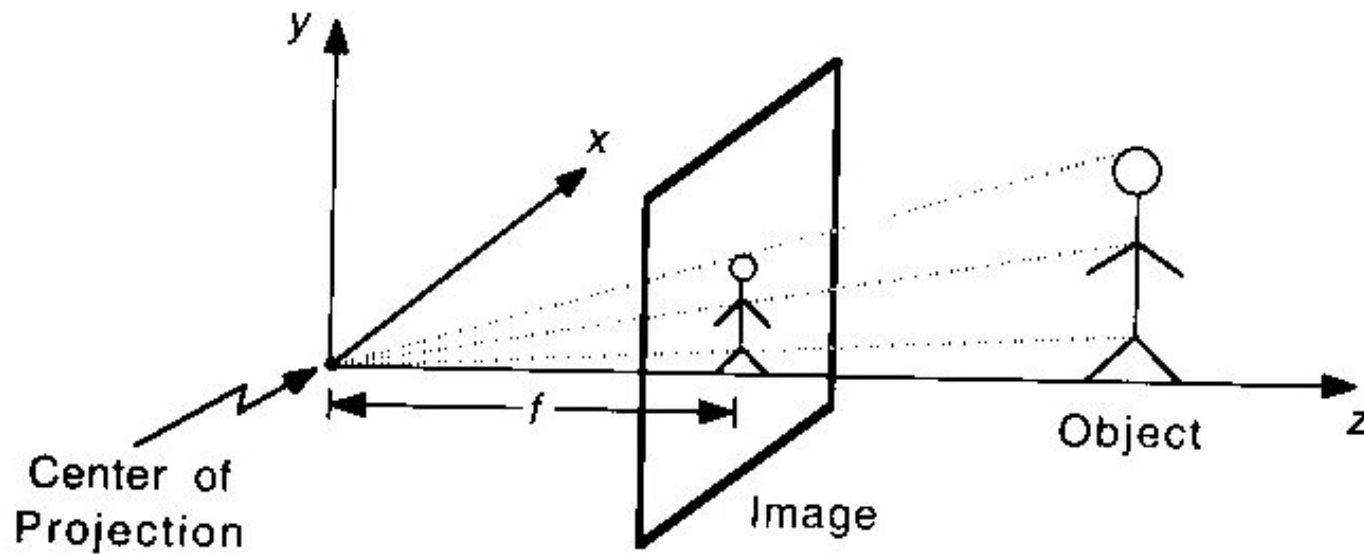


**Basic hypothesis:** Texture resides on the surface and has no thickness.

—> Computation under:

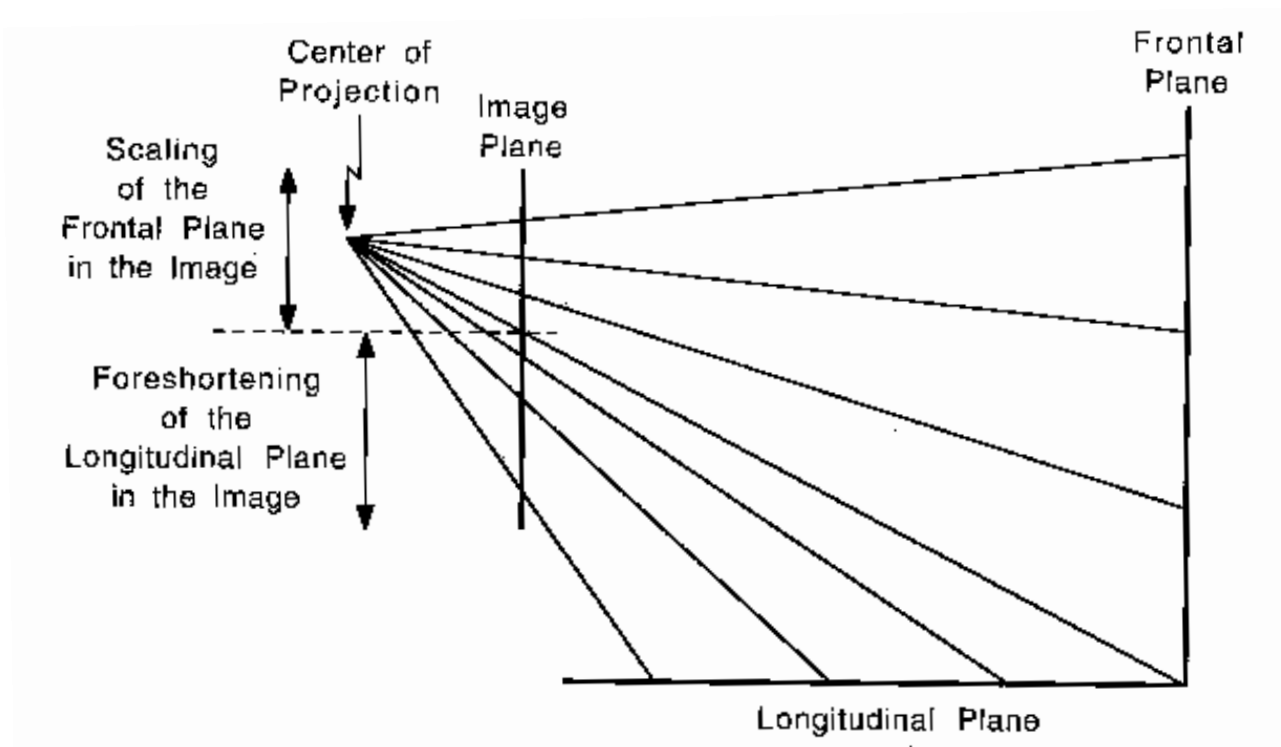
- Perspective projection
- Paraperspective projection
- Orthographic projection

# Reminder: Perspective Projection



$$u = f \frac{x}{z}$$
$$v = f \frac{y}{z}$$

# Perspective Distortion



- The perspective projection distortion of the texture
- depends on both depth and surface orientation,
  - is anisotropic.

# Foreshortening

Depth vs Orientation:

- Infinitesimal vector  $[\Delta x, \Delta y, \Delta z]$  at location  $[x, y, z]$   
image of this vector is

$$\frac{f}{z} \left[ \Delta x - \frac{x}{z} \Delta z, \Delta y - \frac{y}{z} \Delta z \right]$$

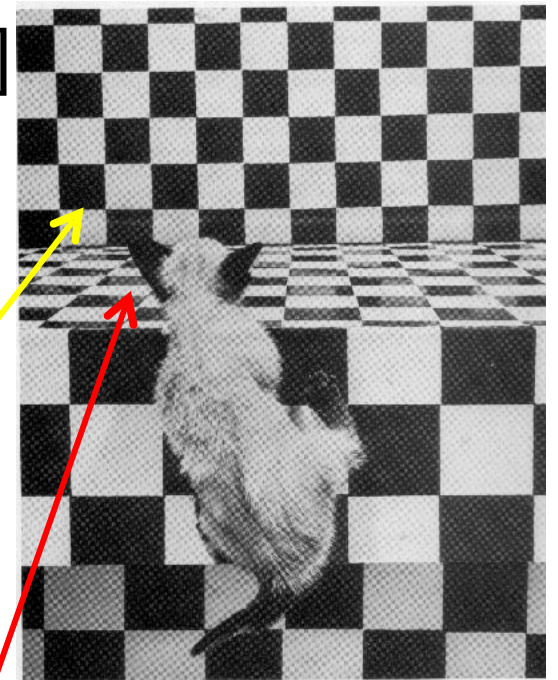
- Two special cases:

- $\Delta z = 0$  :

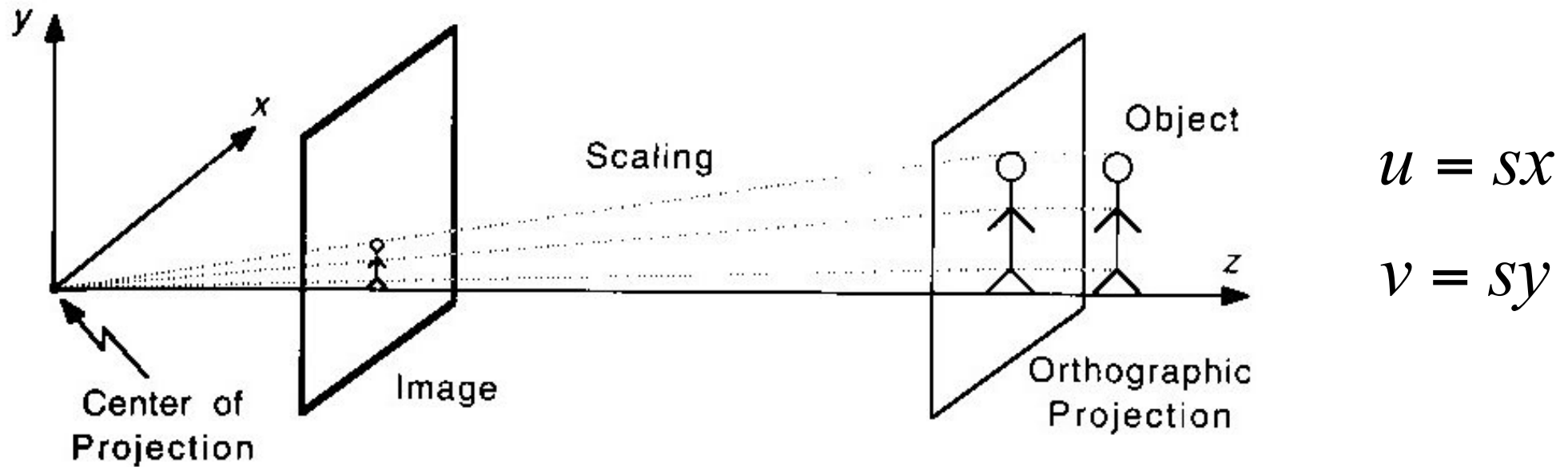
The object is scaled

- $\Delta x = \Delta y = 0$  :

The object is foreshortened



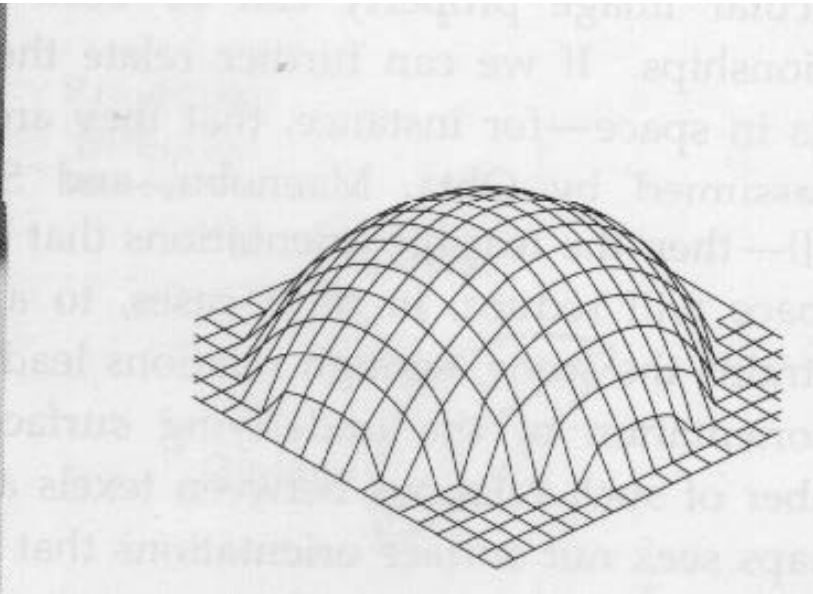
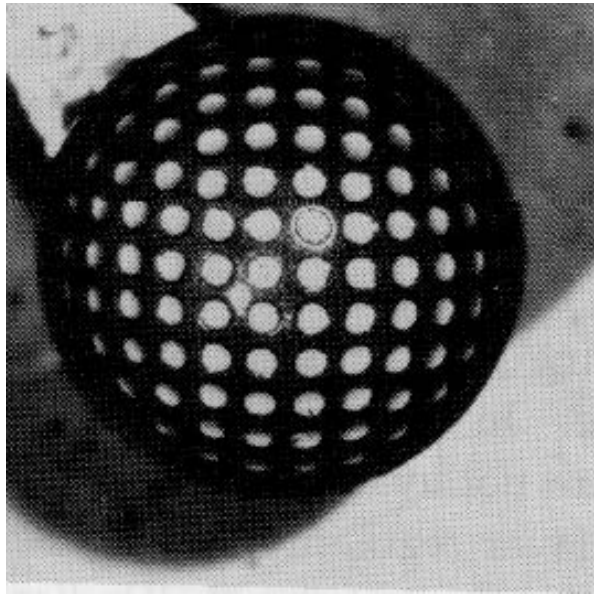
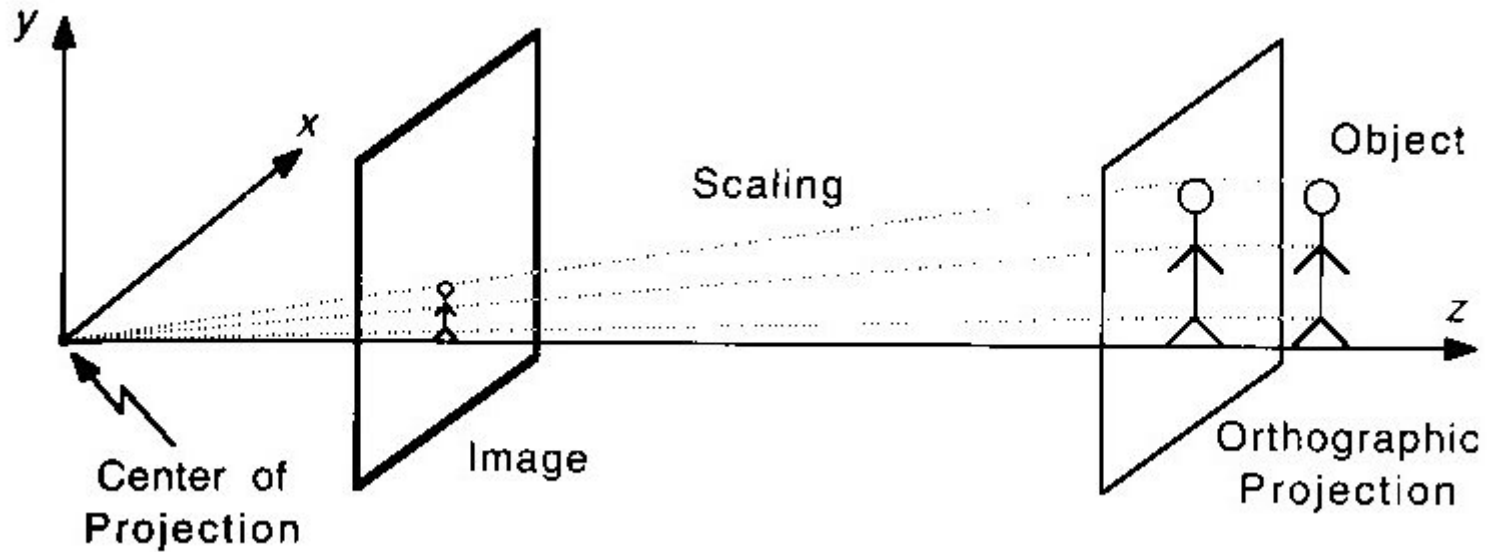
# Reminder: Orthographic Projection



Special case of perspective projection:

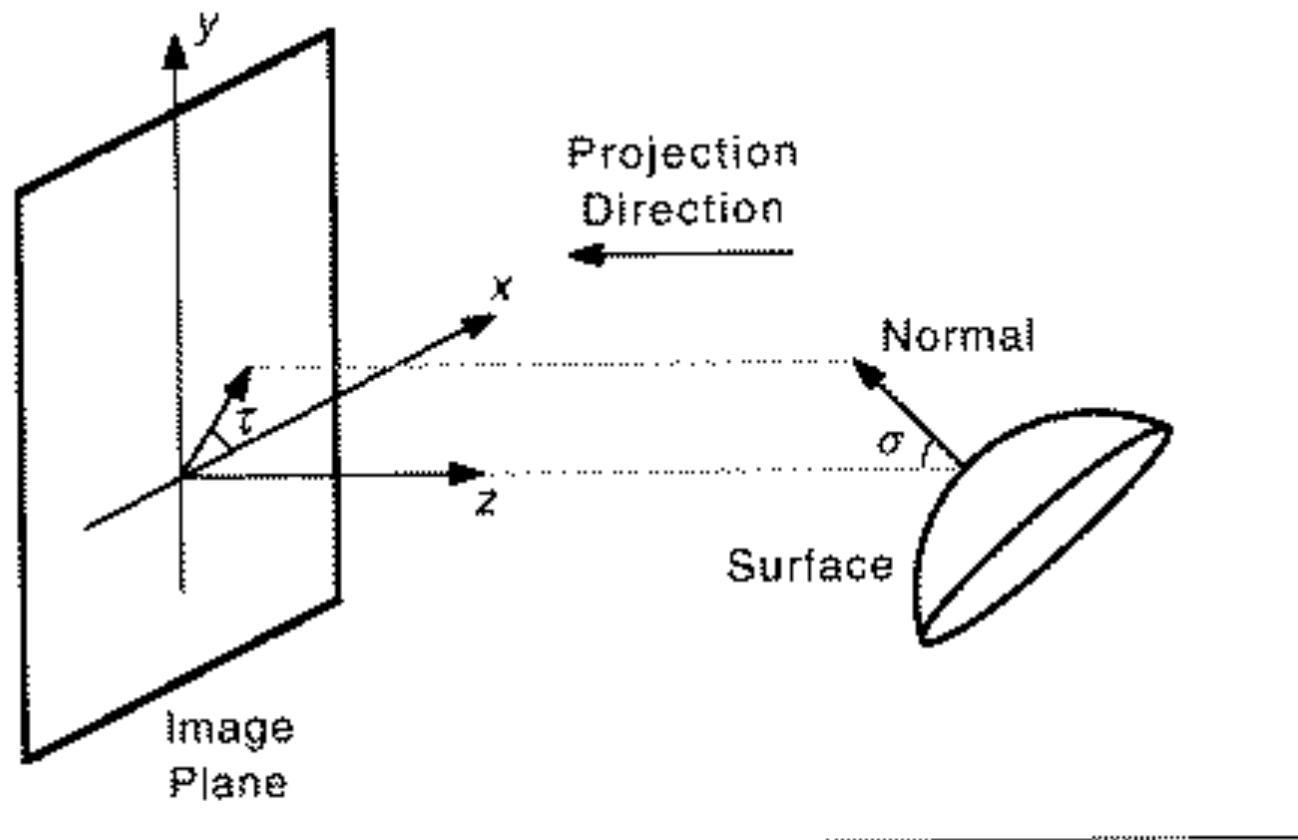
- Large  $f$
  - Objects close to the optical axis
- Parallel lines mapped into parallel lines.

# Orthographic Projection

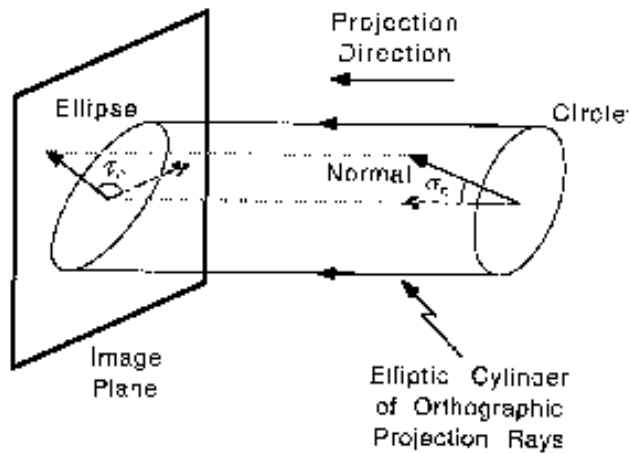




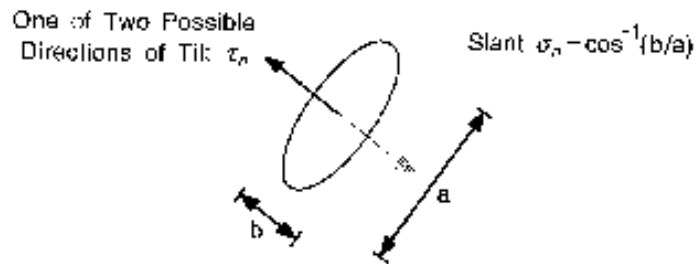
# Tilt And Slant



# Orthographic Projection



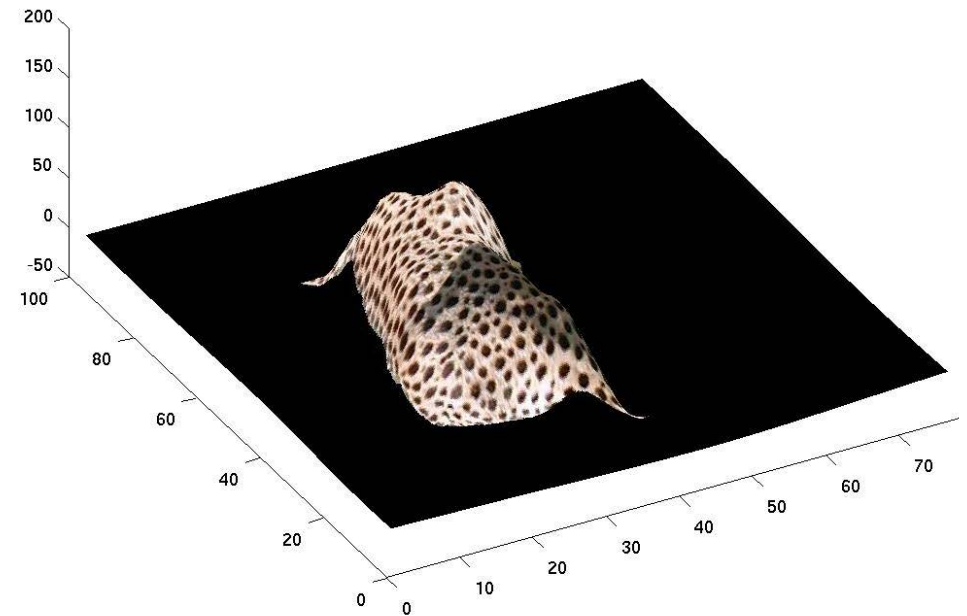
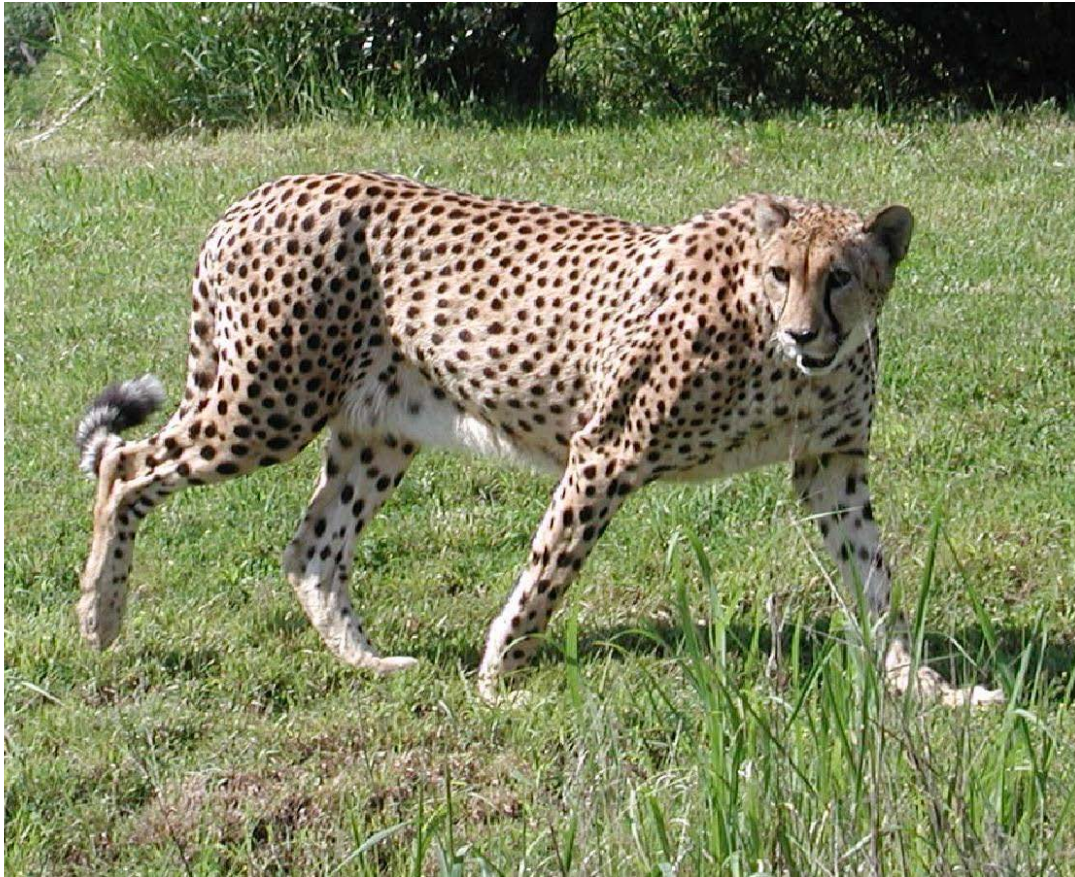
(a)



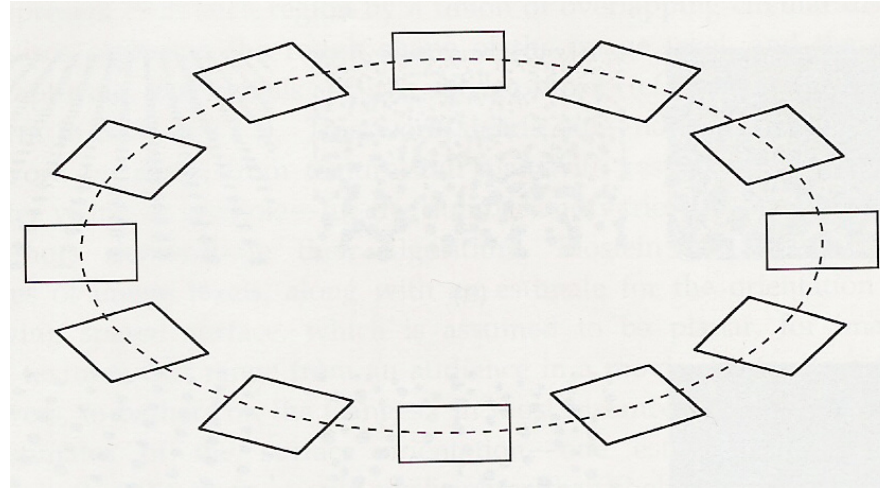
(b)

- **Tilt:** Derived from the image direction in which the surface element undergoes maximum compression.
- **Slant:** Derived from the extent of this compression.

# Cheetah



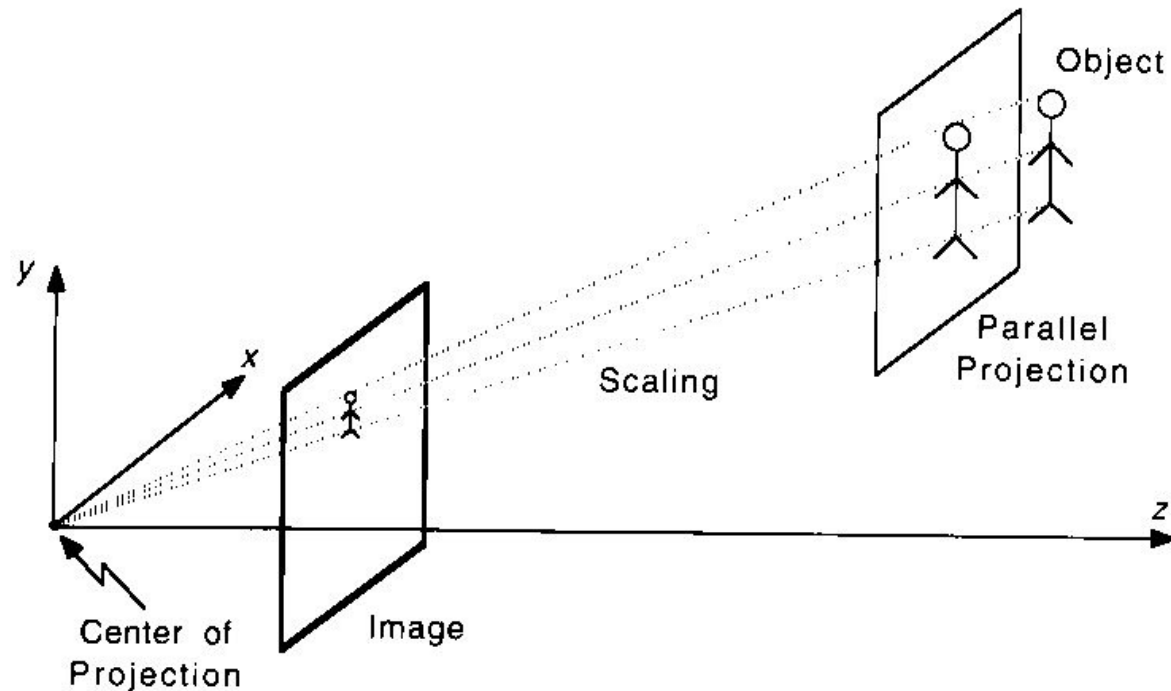
# Perpendicular Lines



Orthographic projections of squares that are rotated with respect to each other in a plane inclined at  $\omega=60^\circ$  to the image plane.

$$\frac{\|\mathbf{p}_1/l_1 \times \mathbf{p}_2/l_2\|}{\|\mathbf{p}_1/l_1\|^2 + \|\mathbf{p}_2/l_2\|^2} = \frac{\cos(\omega)}{1 + \cos^2(\omega)}$$

# Parapespective Projection

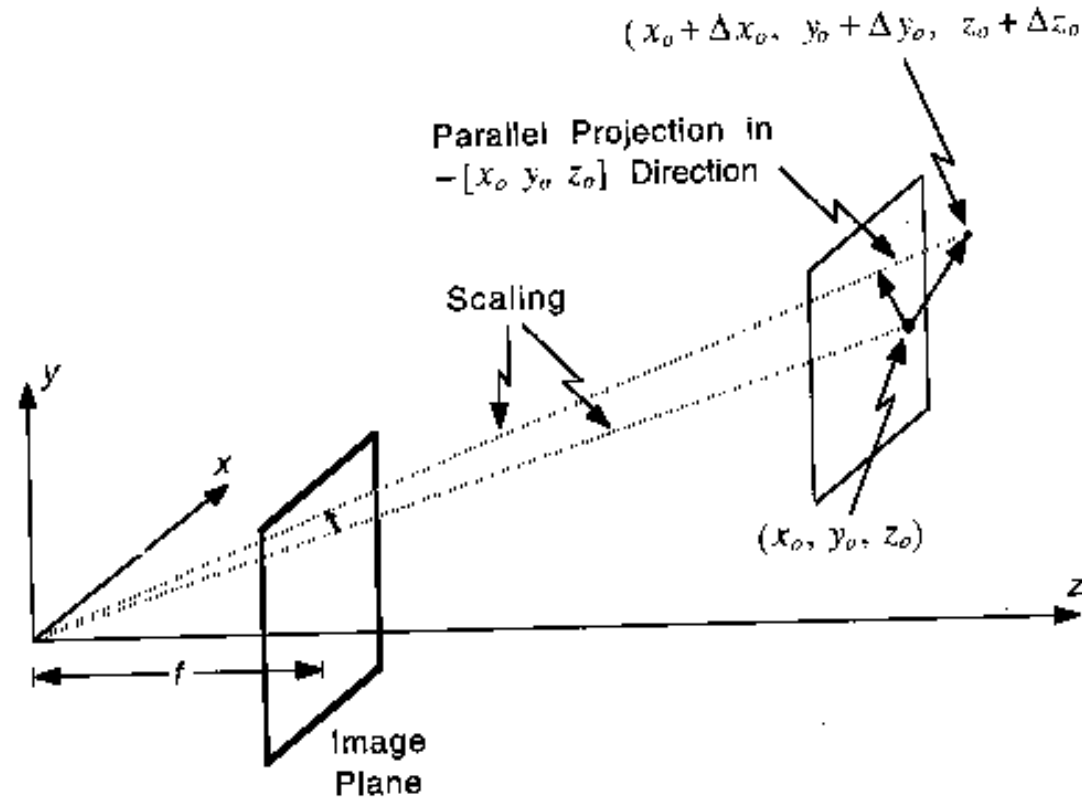


Generalization of the orthographic projection:

- Object dimensions small wrt distance to the center of projection.

→ Parallel projection followed by scaling

# Parapespective Projection

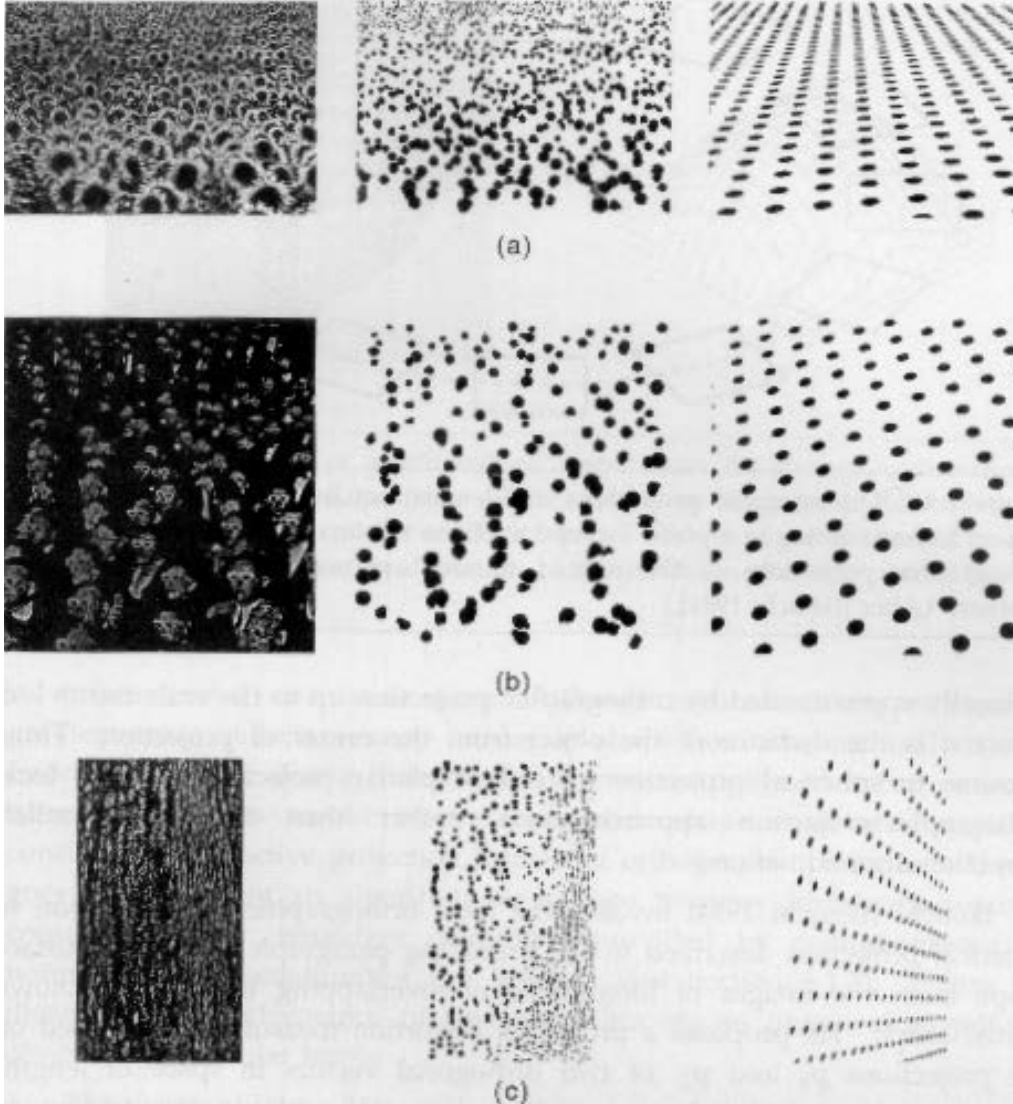


- For planar texels:

$$\text{Projected Area.} \rightarrow A' = -\frac{f^2}{z_0^3} \mathbf{n} \cdot [x_0 \ y_0 \ z_0] A \leftarrow \text{True Area.}$$

Unknown surface normal.

# Paraperspective Projection



## Texels:

- Image regions being brighter or darker than their surroundings.
- Assumed to have the same area in space.

→ Given enough texels, it becomes possible to estimate the normal.

# Texture Gradient





# Statistical Shape Recovery



Measure texture density as opposed to texel area, that is, the number of textural primitives per unit surface.

Assuming the texture to be **homogeneous**, we have:



$$\psi \mathbf{n} \propto \mathbf{b}$$

$$\psi = \begin{bmatrix} u_1 & v_1 & 1 \\ \dots & \dots & \dots \\ u_n & v_n & 1 \end{bmatrix}^t$$

$$\mathbf{b} = [b_1, \dots, b_n]^t$$

$$\Rightarrow \mathbf{n} = \frac{\psi \mathbf{n}}{\|\psi \mathbf{n}\|}$$

Unknown surface normal.

Image coordinates.

Function of density.

# Strengths and Limitations (2015)

## Strengths:

- Emulates an important human ability.

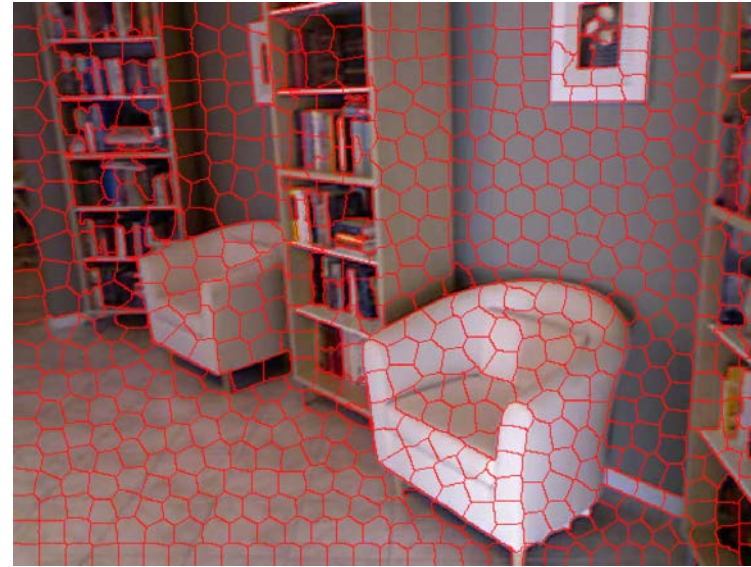
## Limitations:

- Involves very strong assumptions.
- Only useful in very specific settings.

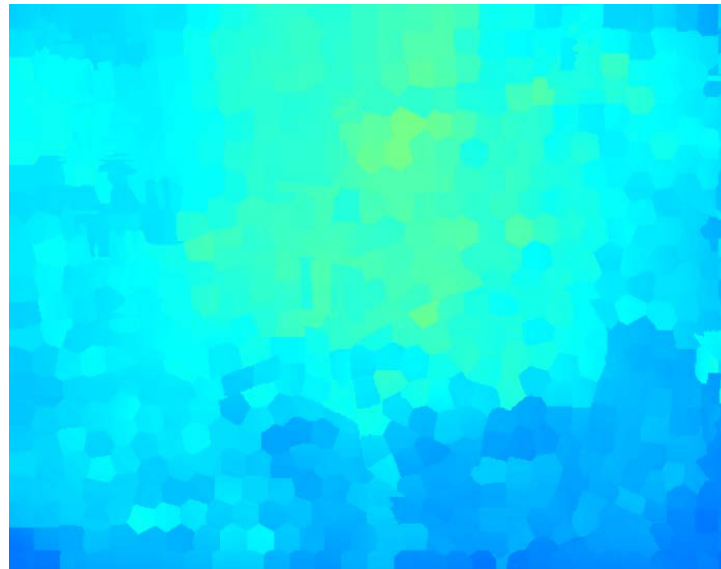
# Machine Learning



Input Image



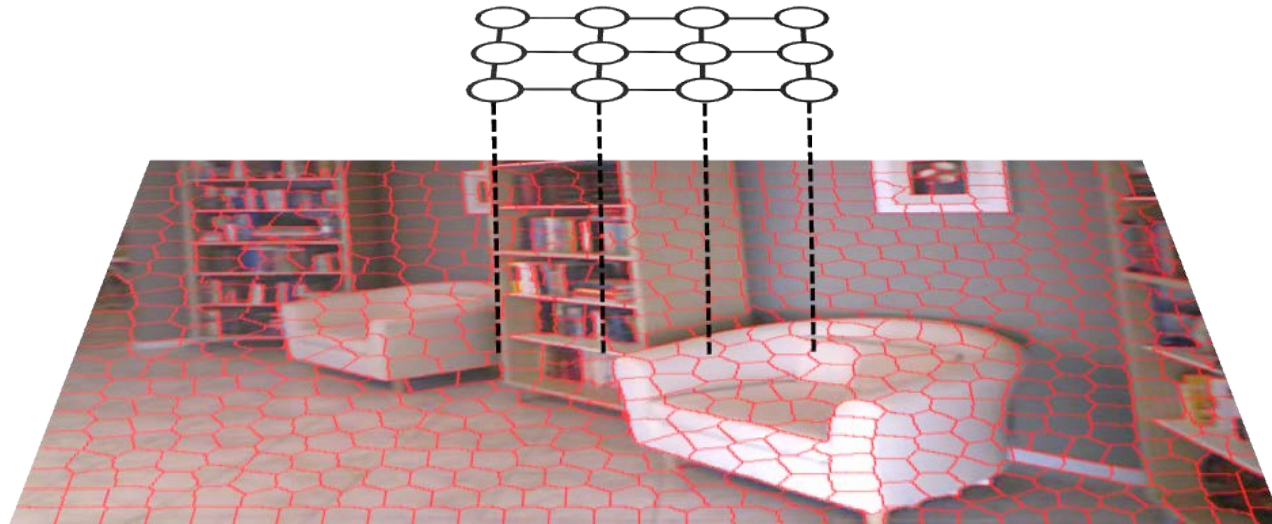
Superpixels



**EPFL** Train a regressor to predict depth  $\rightarrow$  Noisy predictions

# Markov Random Field (MRF)

Graph with vertices and edges



Assign values to the nodes to minimize

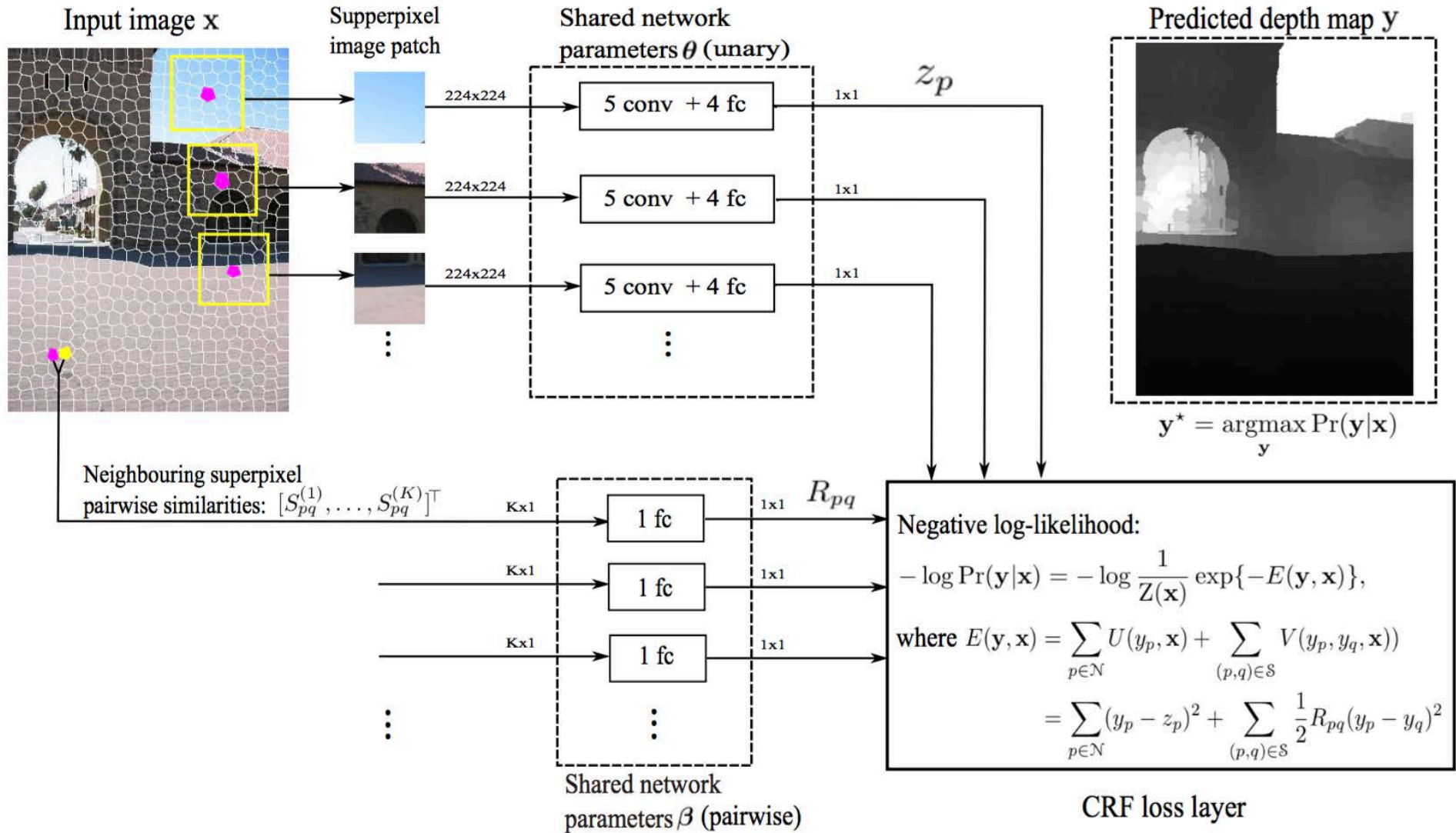
$$E(Y) = \sum_i \varphi(y_i) + \sum_{(i,j)} \psi(y_i, y_j)$$

unary

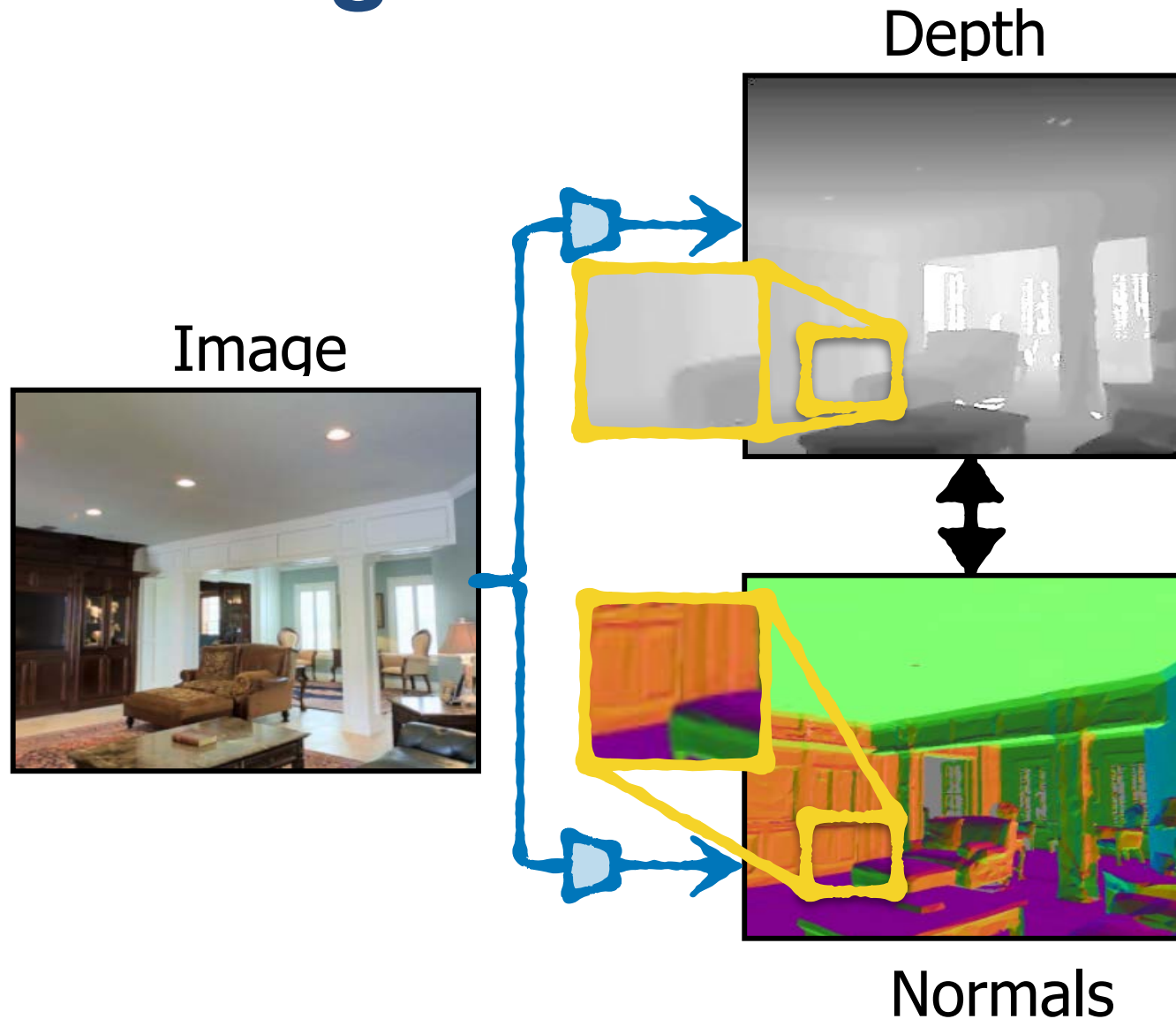
pairwise

—> Enforces consistency

# Deep Learning with MRF

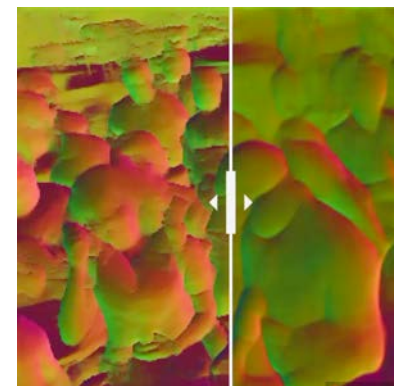
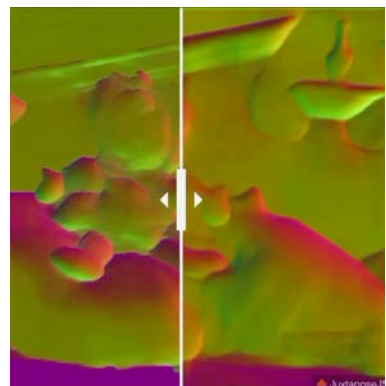
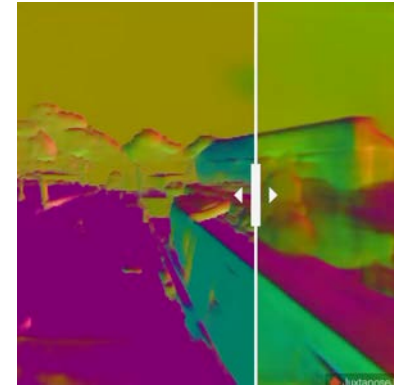
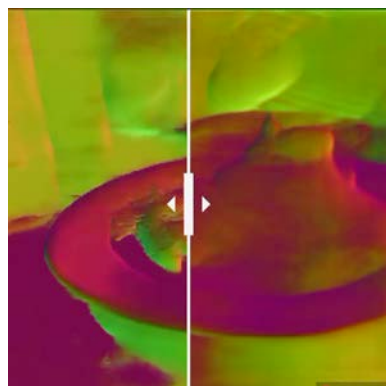
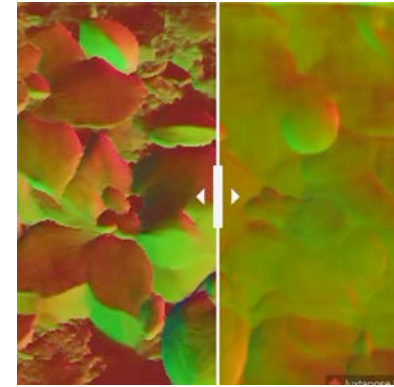
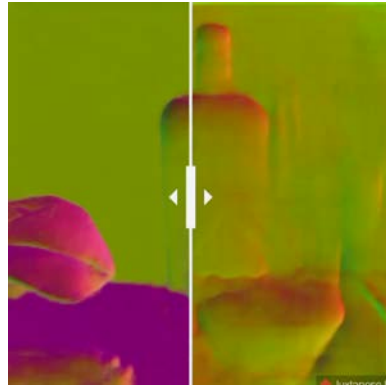


# Enforcing Task Consistency

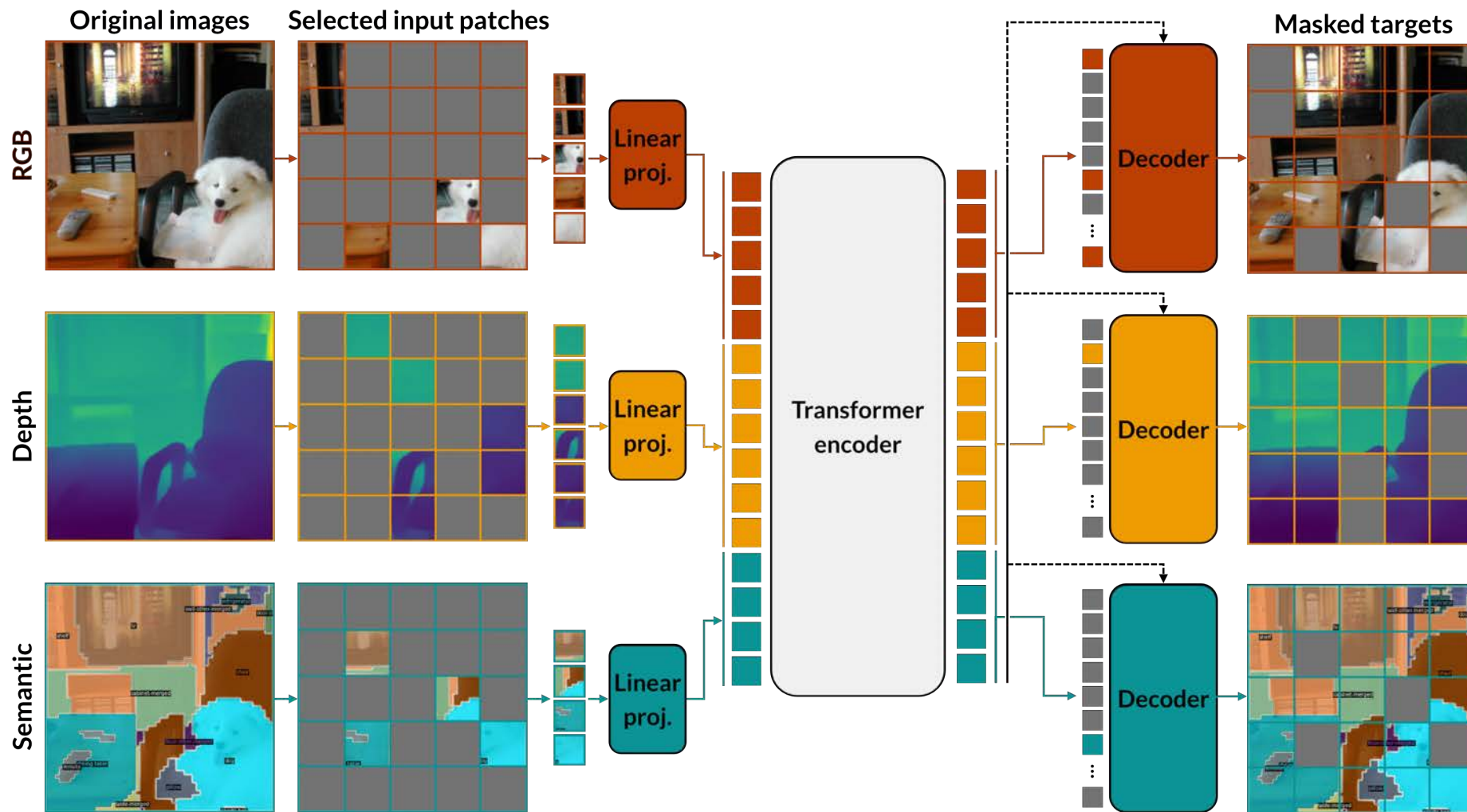


- A network can be trained to predict multiple things.
- Forcing consistency across tasks increases robustness.

# A Very Diverse Training Database Helps

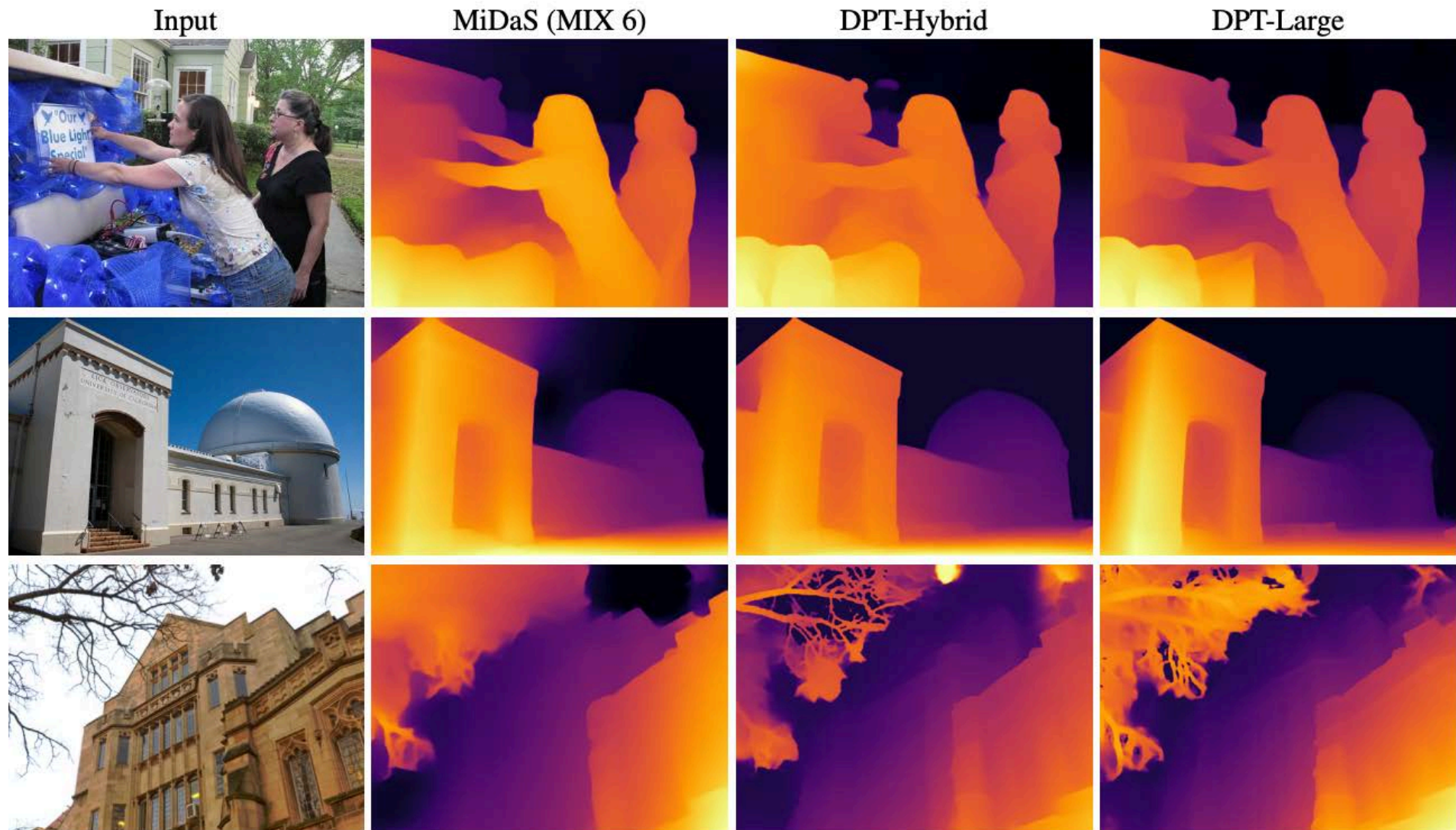


# .. and so does a Transformer Architecture



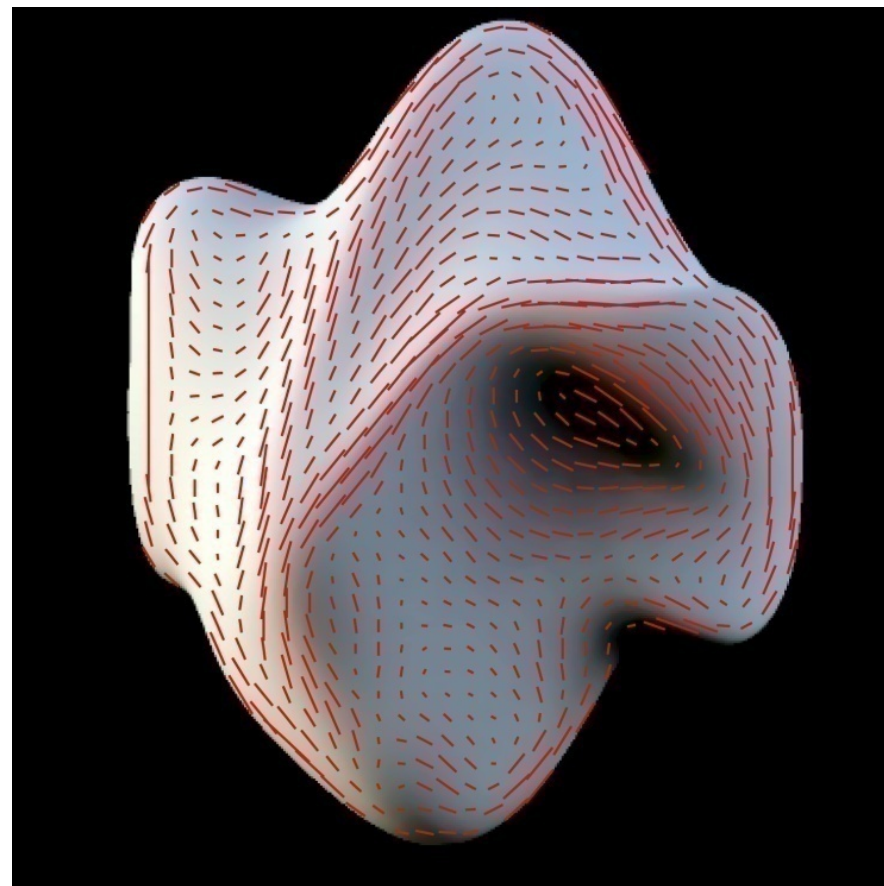
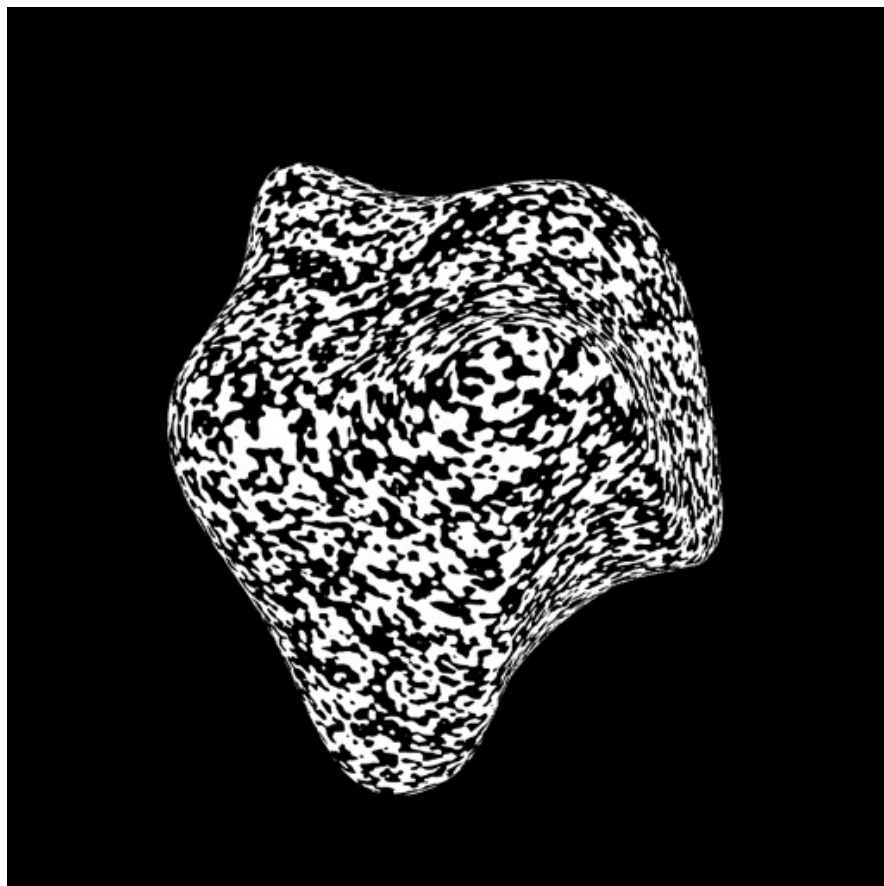


# Using Transformers



- Pros: Good at modeling long range relationships.
- Cons: Flattening the patches loses some amount of information.

# Optional: Illusory Shape Distorsion

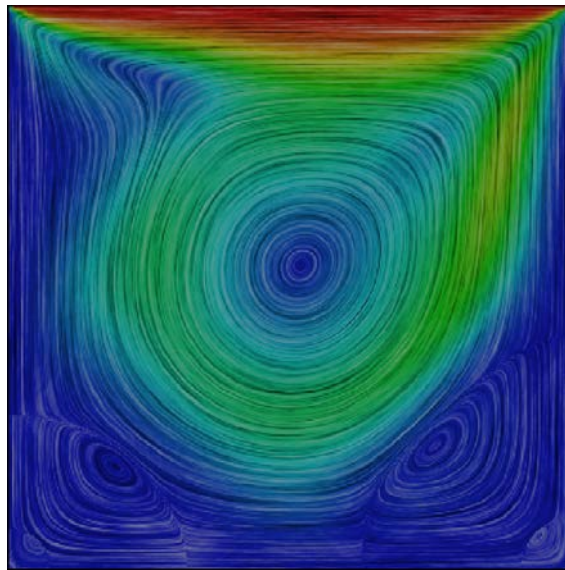


People seem to be sensitive to orientation fields in the cases of both texture and shading.

# Optional: Shape from Smear

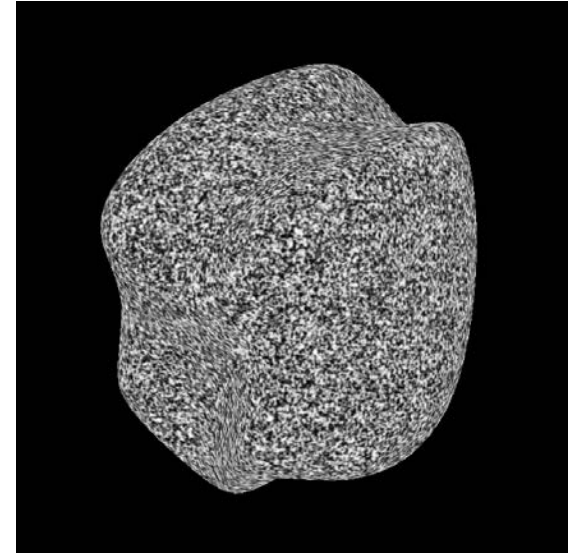
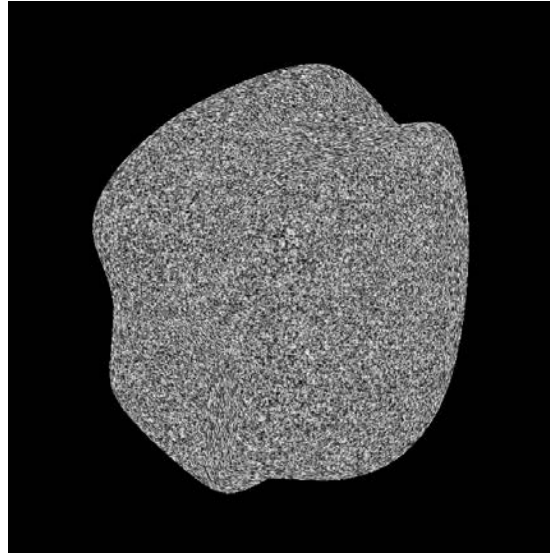
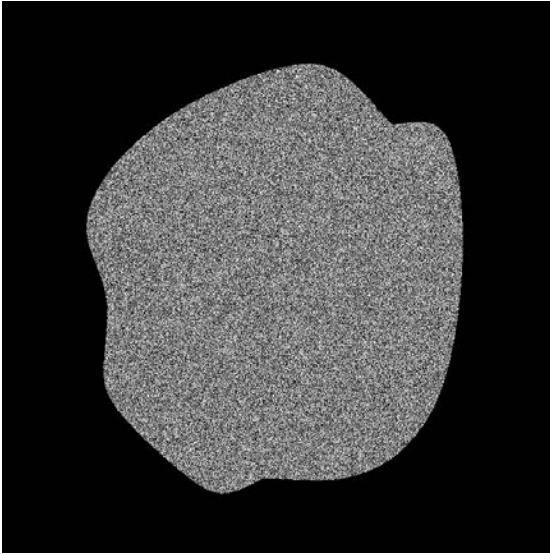
**Hypothesis:** If orientation and scale fields are the key source of information for 3D shape perception, it should be possible to induce a vivid sense of 3D shape by creating 2D patterns with appropriate scale and orientation fields.

**Test:** Use a technique known as Line Integral Convolution to smear the texture along specific orientations and scale appropriately.

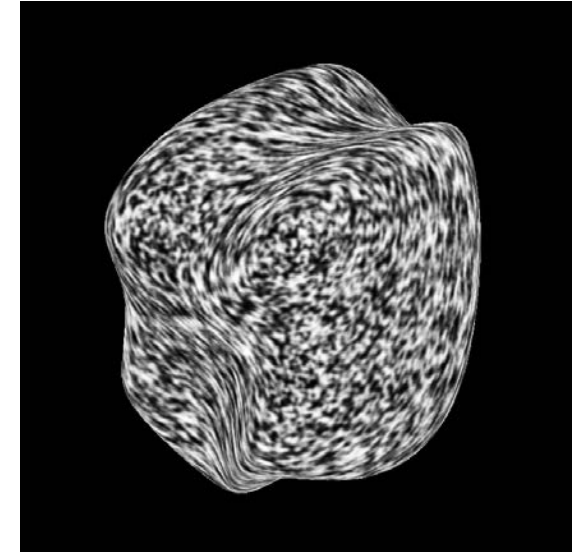
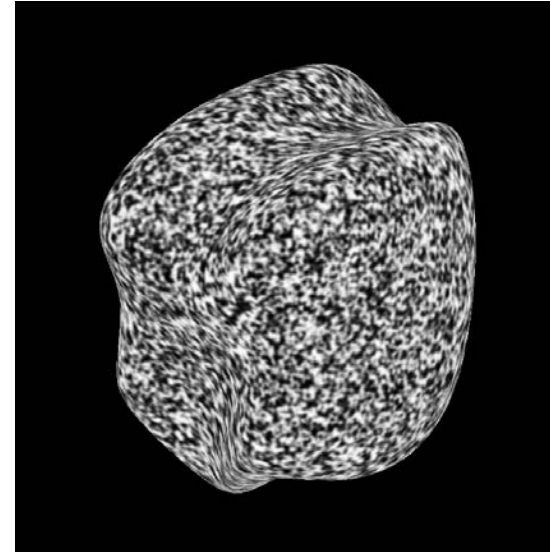
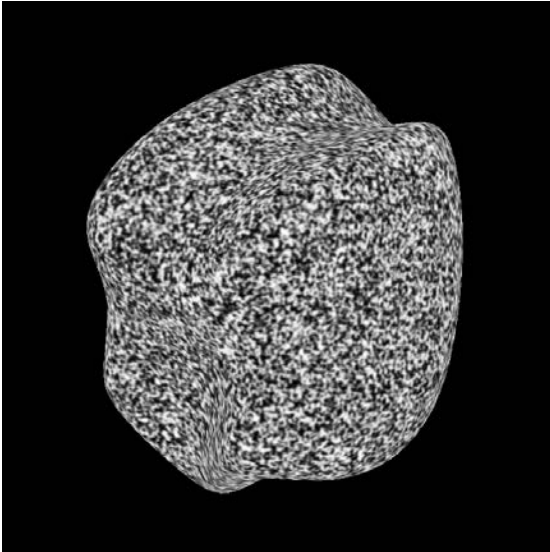


# Optional: Scaling and Smearing

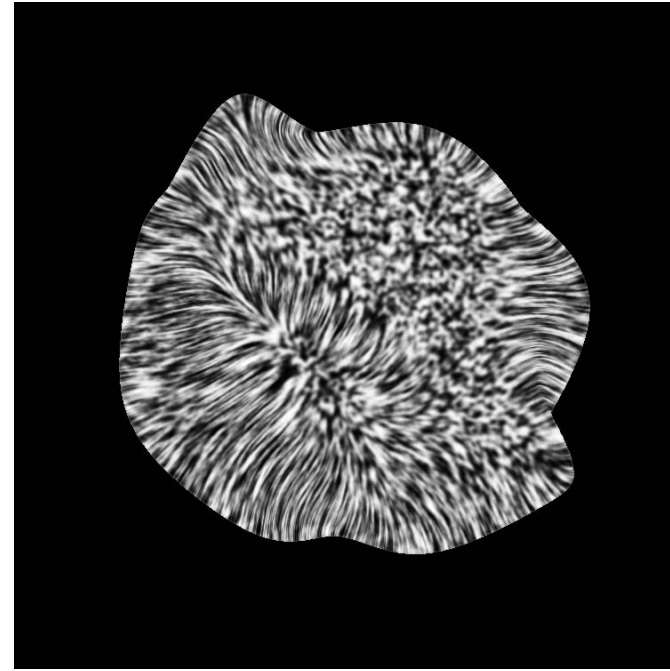
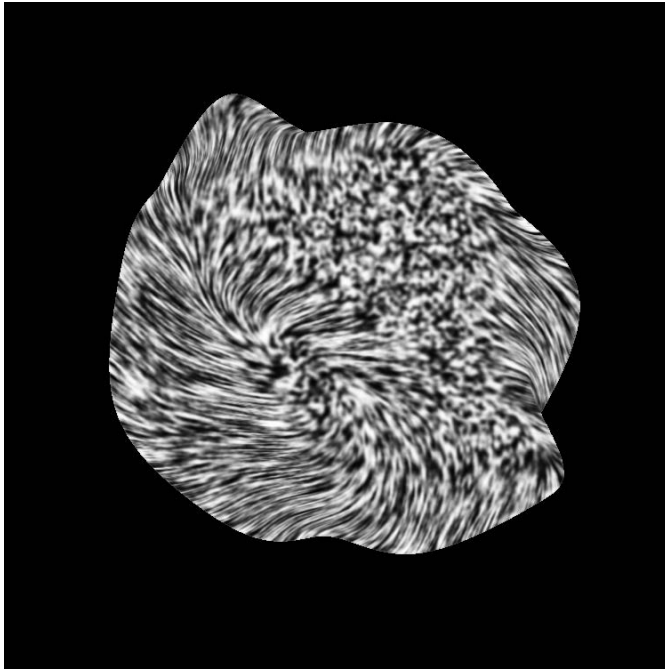
Scaling:



Smearing:



# Optional: Inconsistent Stimulus



The orientation field cannot be integrated

- No depth perception.
- Do we integrate in our heads?
- Is this what the deep nets learn to do?

# Strengths and Limitations

## Strengths:

- Emulates an important human ability.

## Limitations:

- Requires regular texture.
- Involves very strong assumptions.
- Deep learning can be used to weaken them.