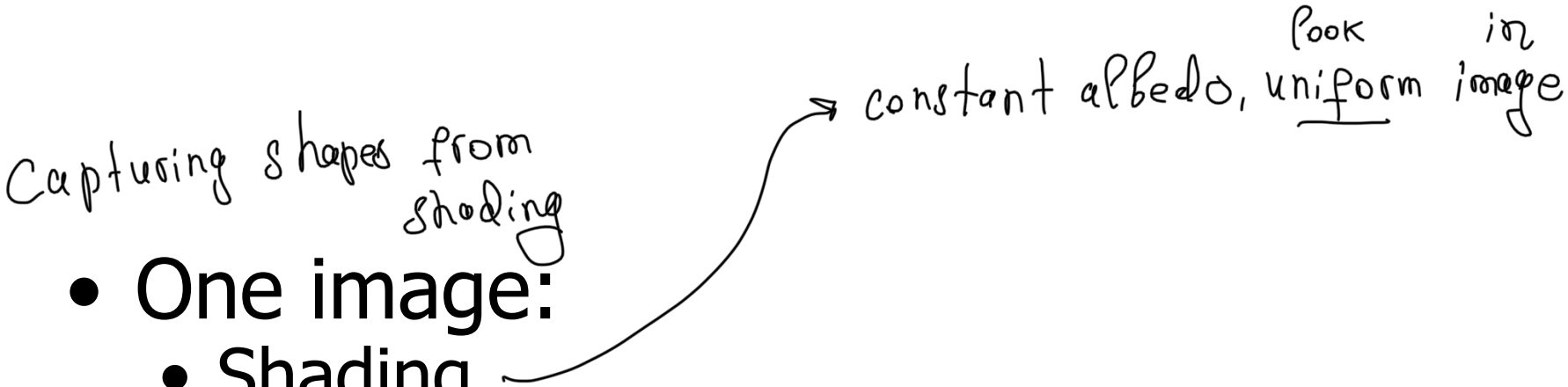
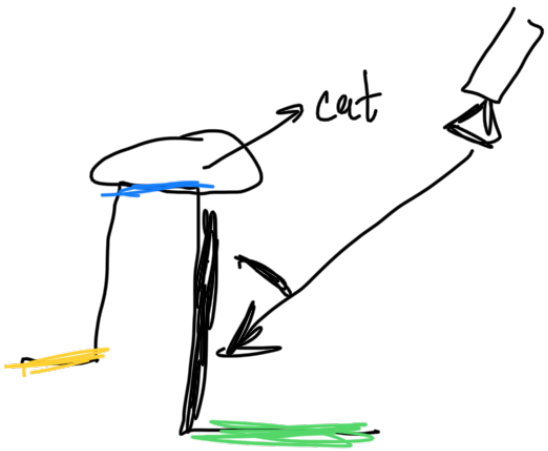
Shape from X



- One image:
  - Shading
  - **Texture**

- Two images or more:
  - Stereo
  - Contours
  - Motion

Shape From X

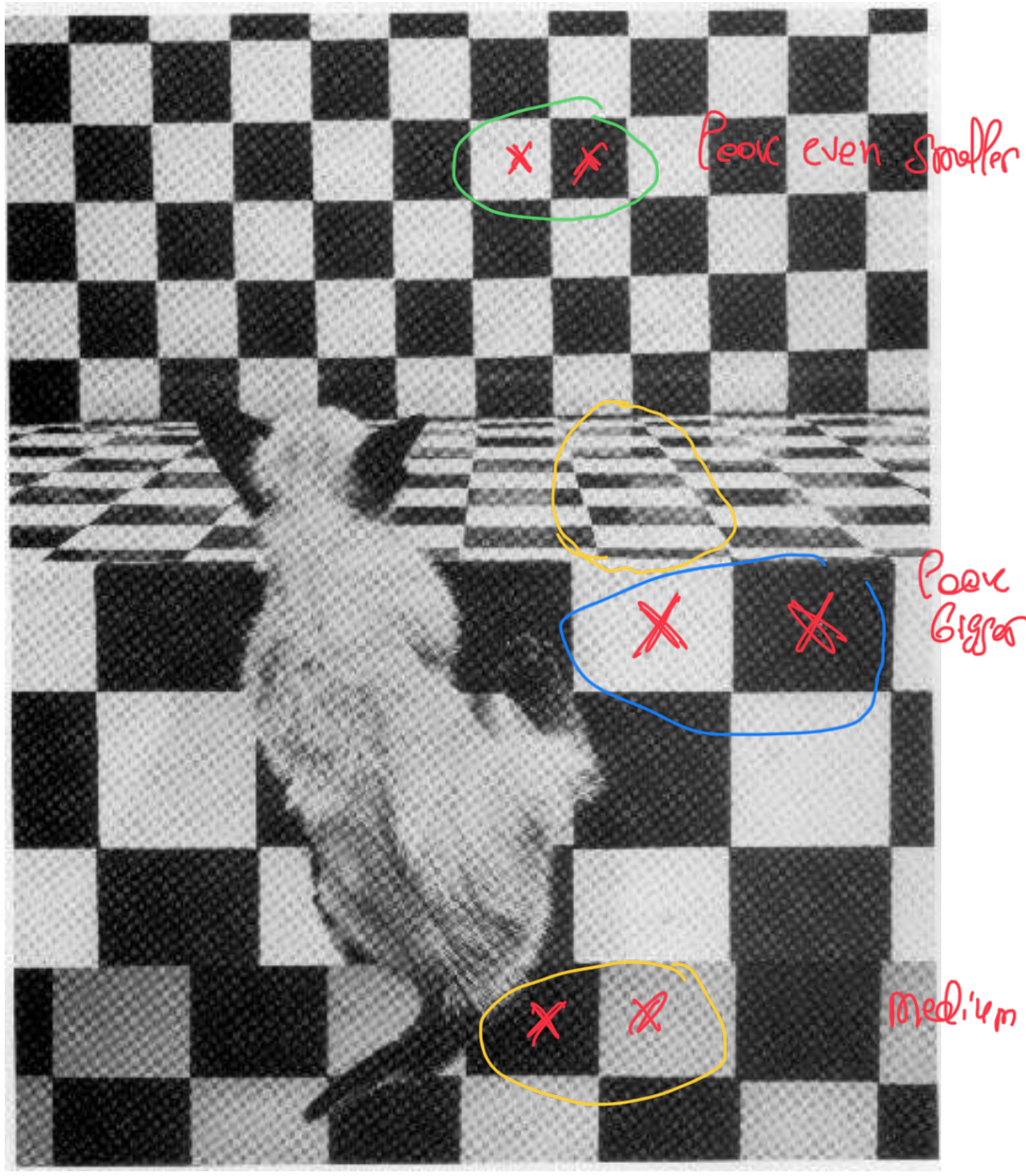


- One image:
  - Shading
  - **Texture**

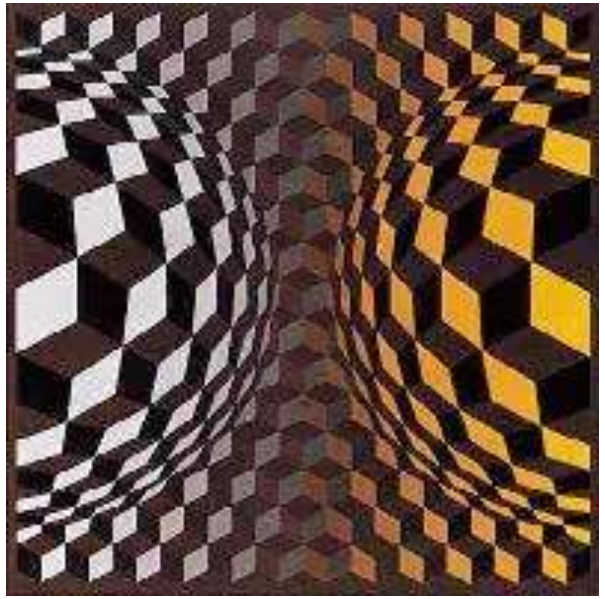
They are not squares in proj ⇒ distorted, foreshortened

- Two images or more:
  - Stereo
  - Contours
  - Motion

In perspective projective, things that are further look smaller



Shape From Texture



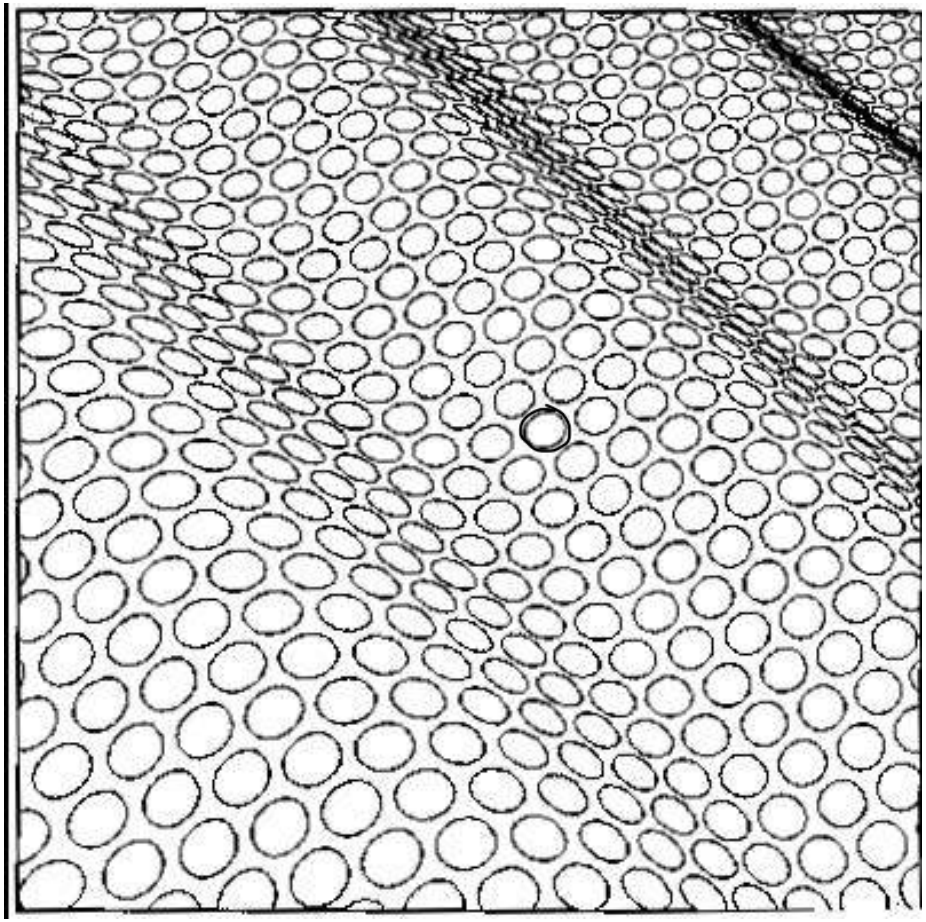
Recover surface orientation or surface shape from image texture:

- 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.

distance also

Structural Shape Recovery

projection deforms them → How they are deformed
projected circles ✓ tells us sth about their shapes

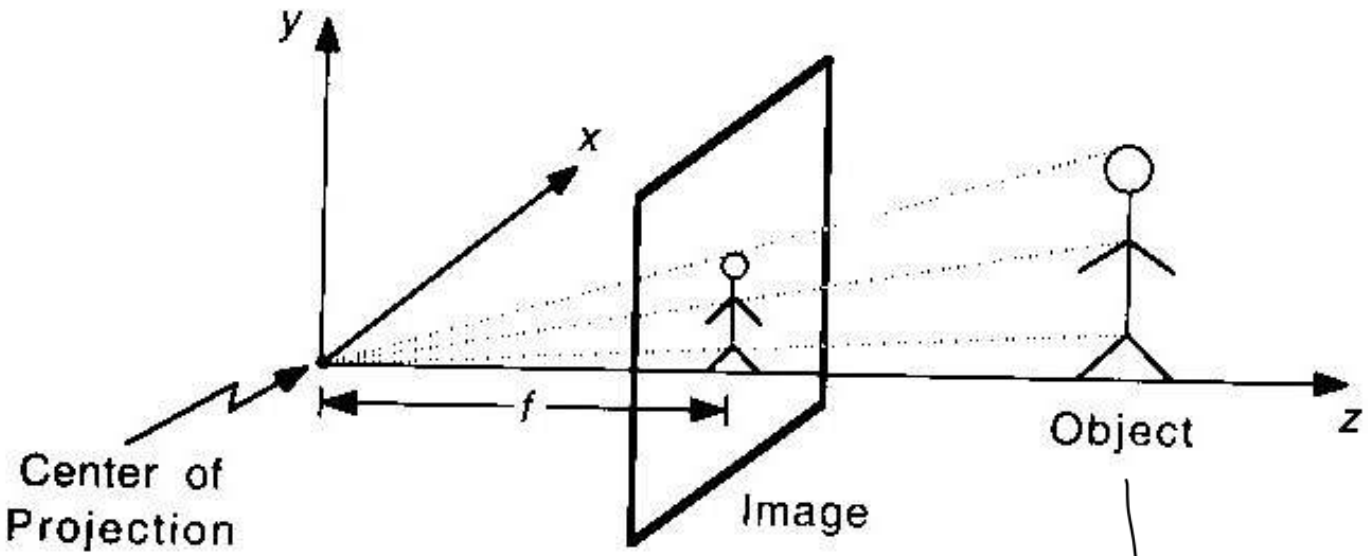


Basic hypothesis: Texture resides on the surface and has no thickness.

—> Computation under:

- Perspective projection
- Paraperspective projection
- Orthographic projection

Reminder: Perspective Projection

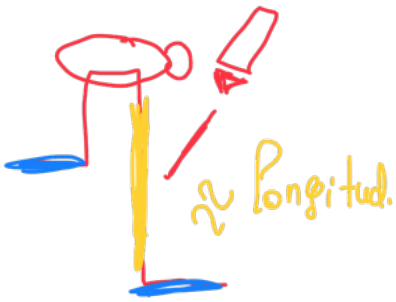


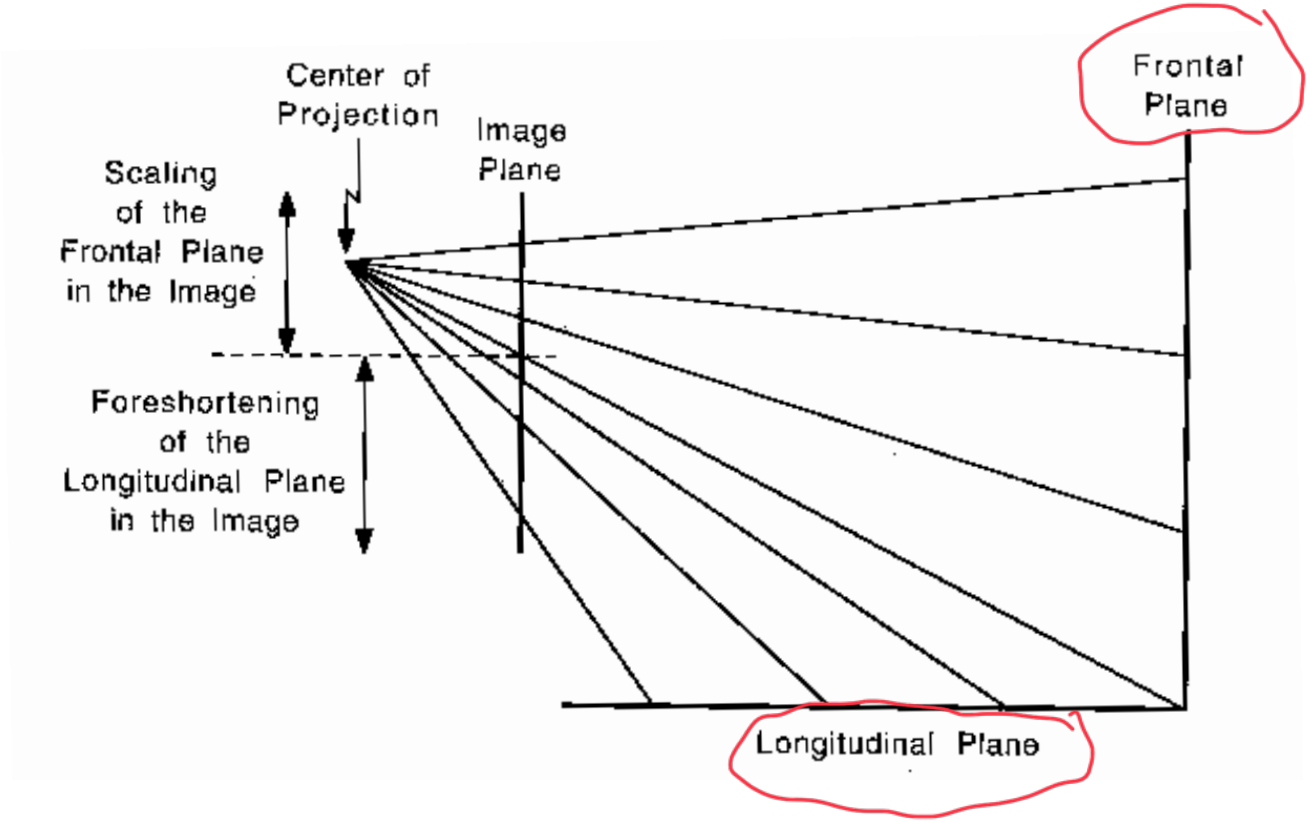
$$u = f\frac{x}{z}$$

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

Camera coordinates

Perspective Distortion





≈ frontal plane

The perspective projection distortion of the texture

- depends on both depth and surface orientation,
- is anisotropic.

tangent surface and position of the camera

Foreshortening

distance from camera → point in the scene

angle at which surface is oriented relative to camera

Depth vs Orientation:

- Infinitesimal vector [Δx,Δy,Δz] at location [x,y,z] image of this vector is (according to dynamics eq'n)

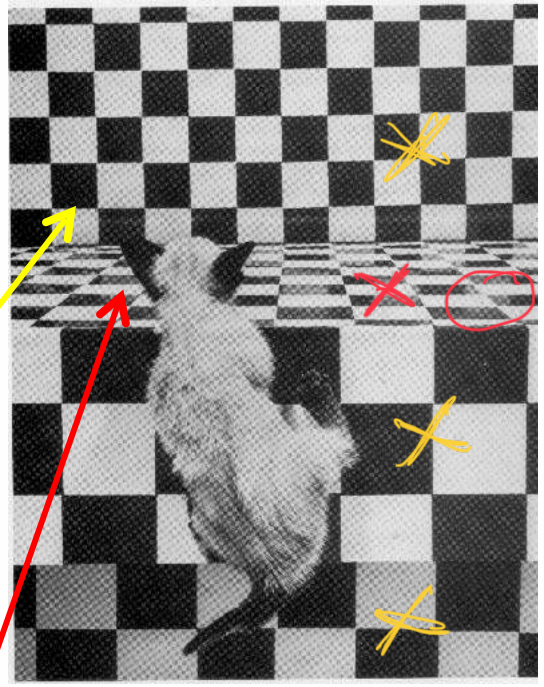
$$\frac{f}{z}\left[\Delta x - \frac{x}{z}\Delta z, \Delta y - \frac{y}{z}\Delta z\right]$$

projection of Δx, Δy, Δz

occur in depth changes do not

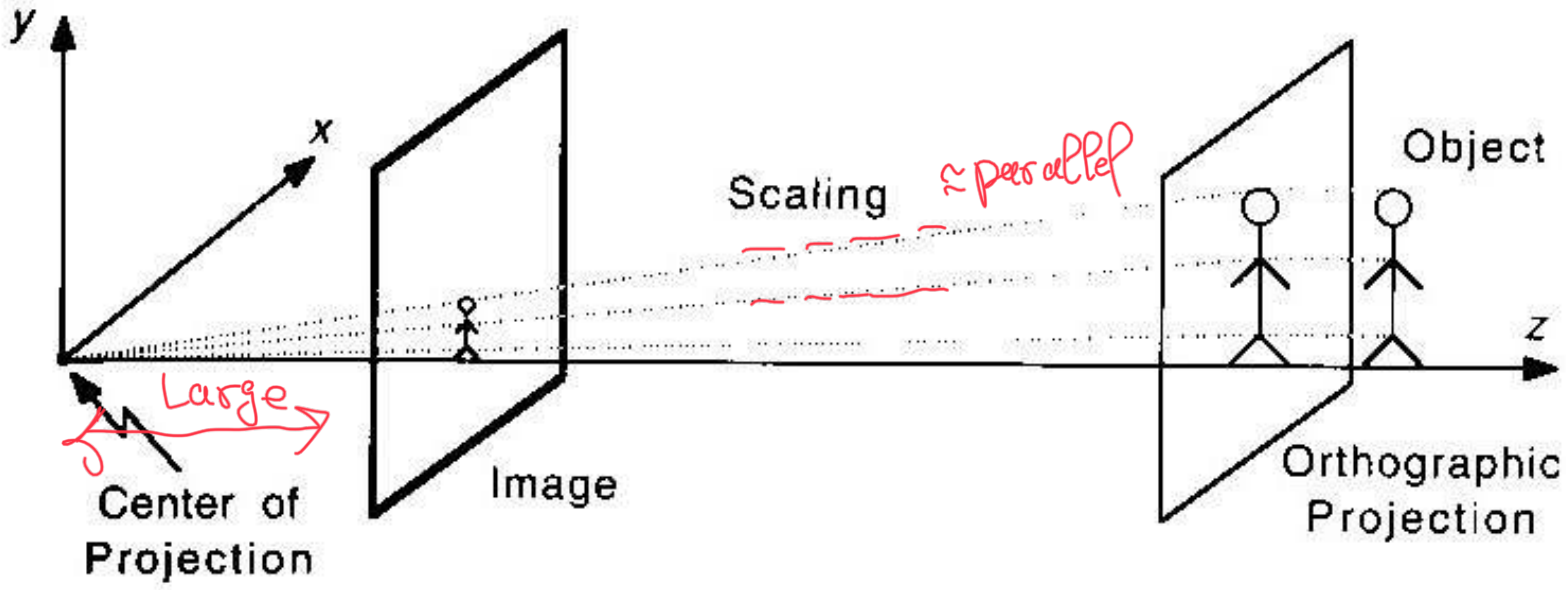
- Two special cases:
  - Δz=0 : The object is scaled (frontal plane)
  - Δx=Δy=0 : The object is foreshortened (longitudinal plane)

(division by z^2)



scrunched more in 1 direction than other

Reminder: Orthographic Projection



$$u = sx$$

$$v = sy$$

→ Does not account for the fact that objects appear smaller when they are further away

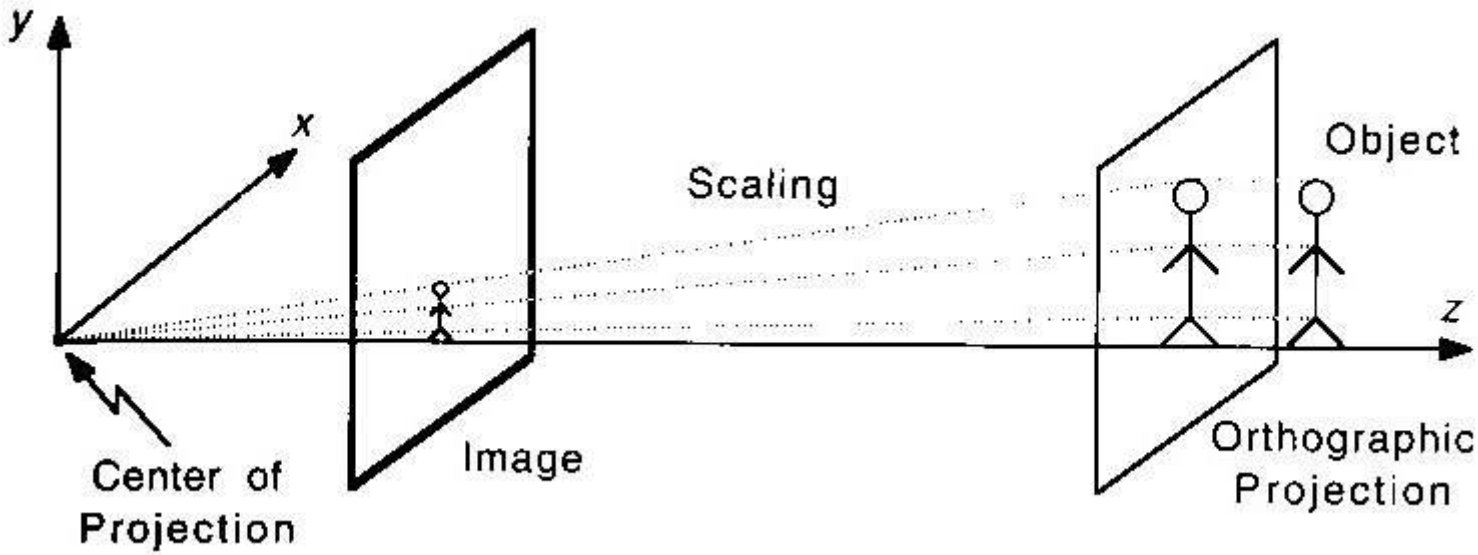
Special case of perspective projection:

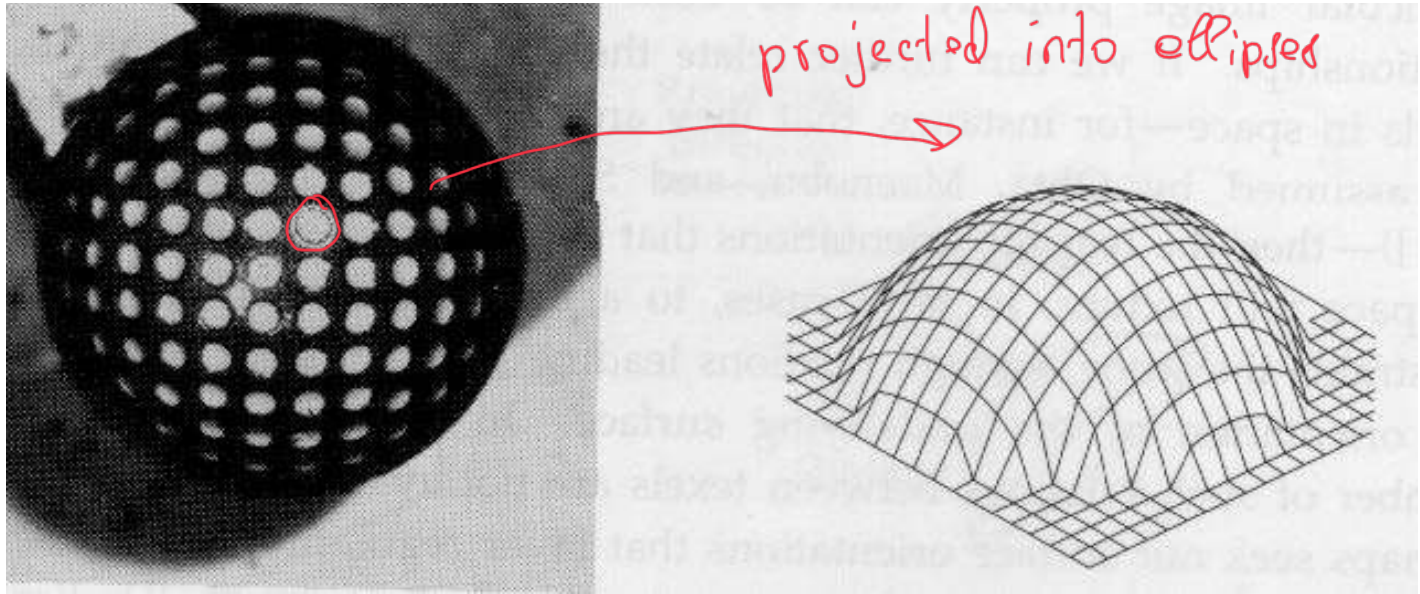
- Large f
- Objects close to the optical axis

→Parallel lines mapped into parallel lines.

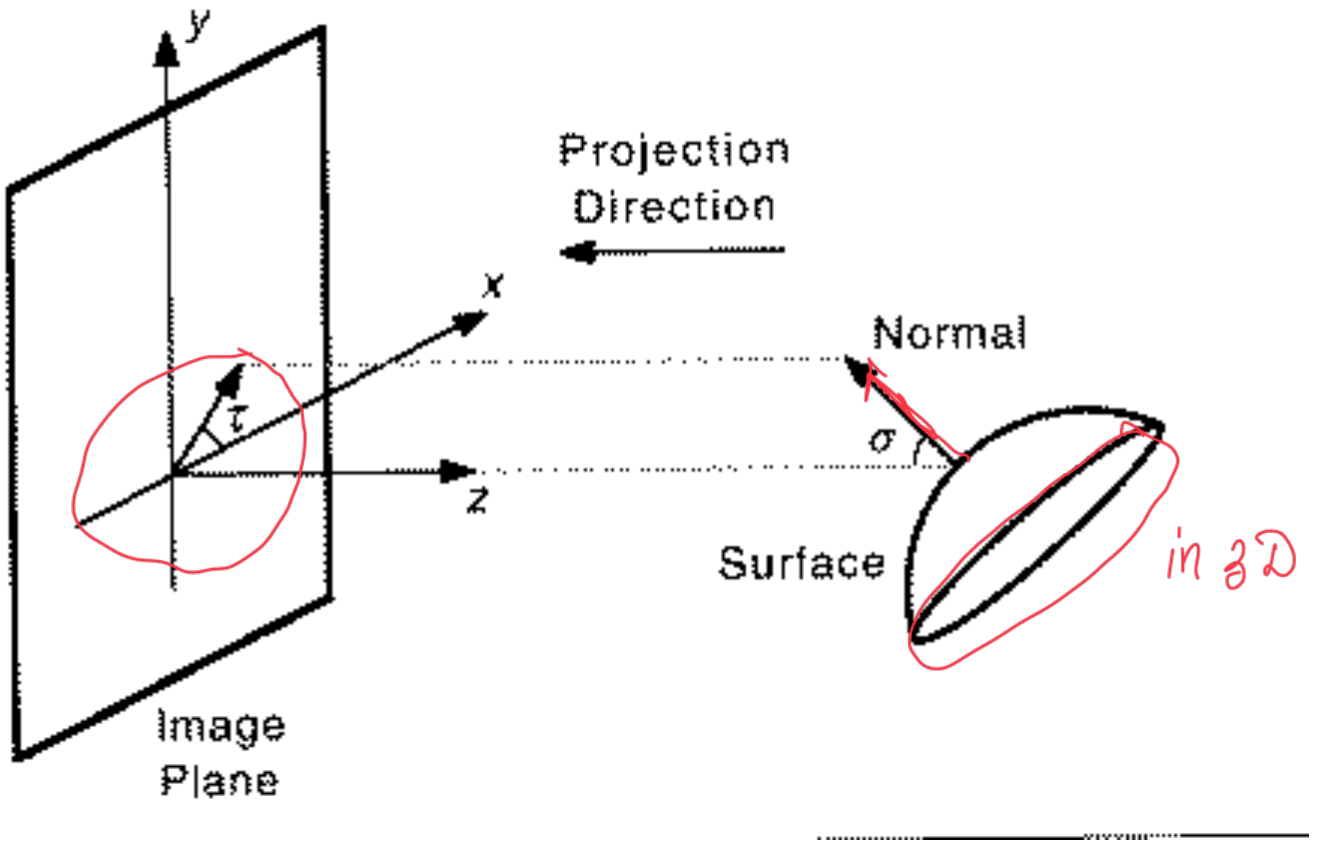
No foreshortening (distortion that makes an object appear shorter than it's due to angled view)

Orthographic Projection





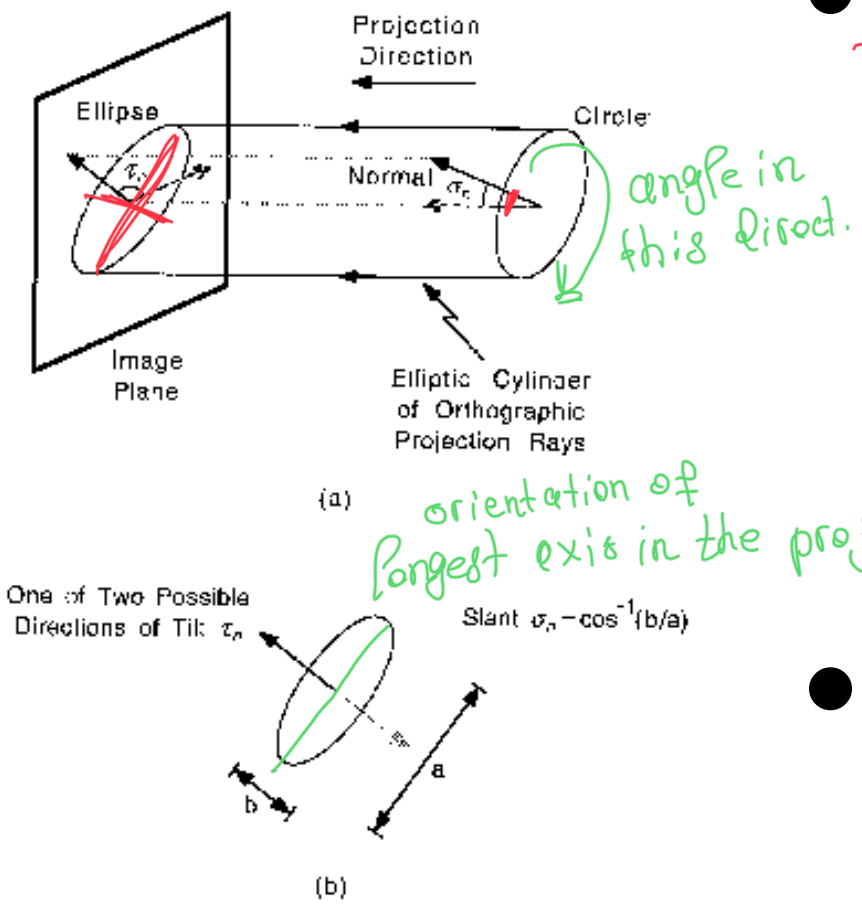
Tilt And Slant



Orthographic Projection

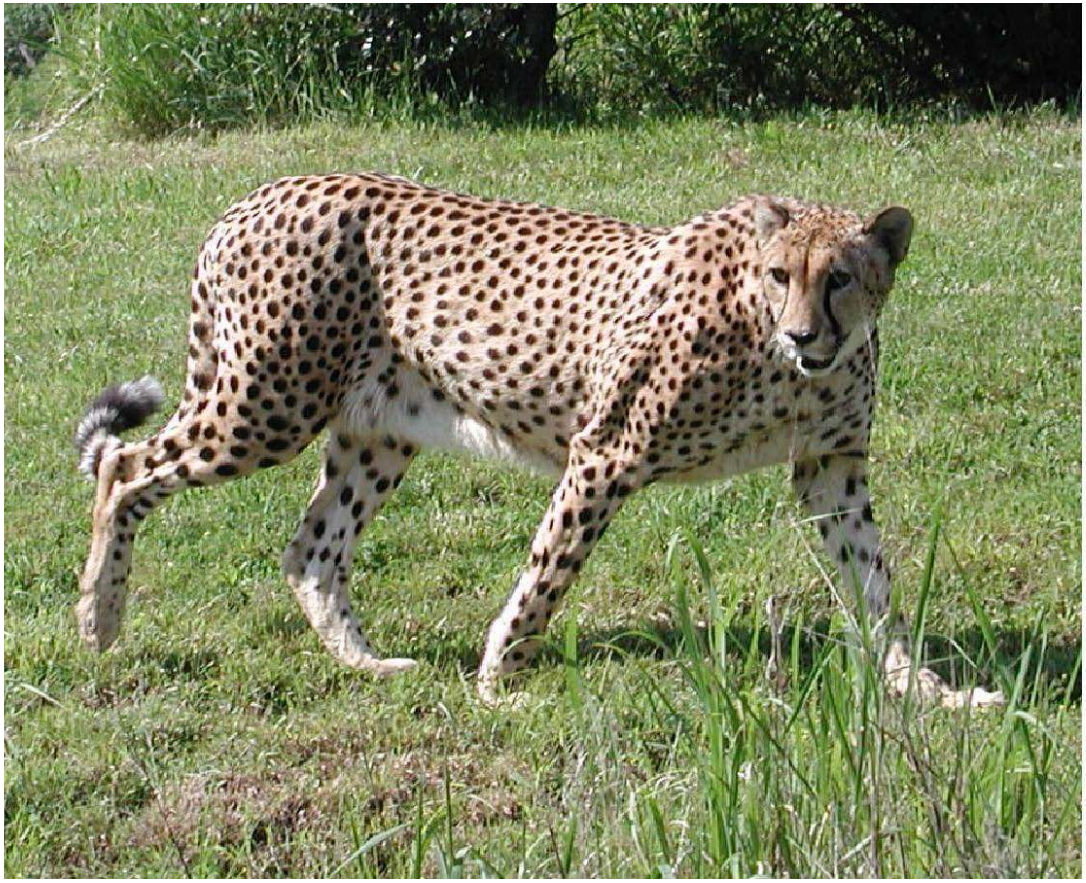
$\alpha = 0 \Rightarrow$ projecting into circle
$\uparrow \alpha \Rightarrow$ more elongated ellipse / ellipse

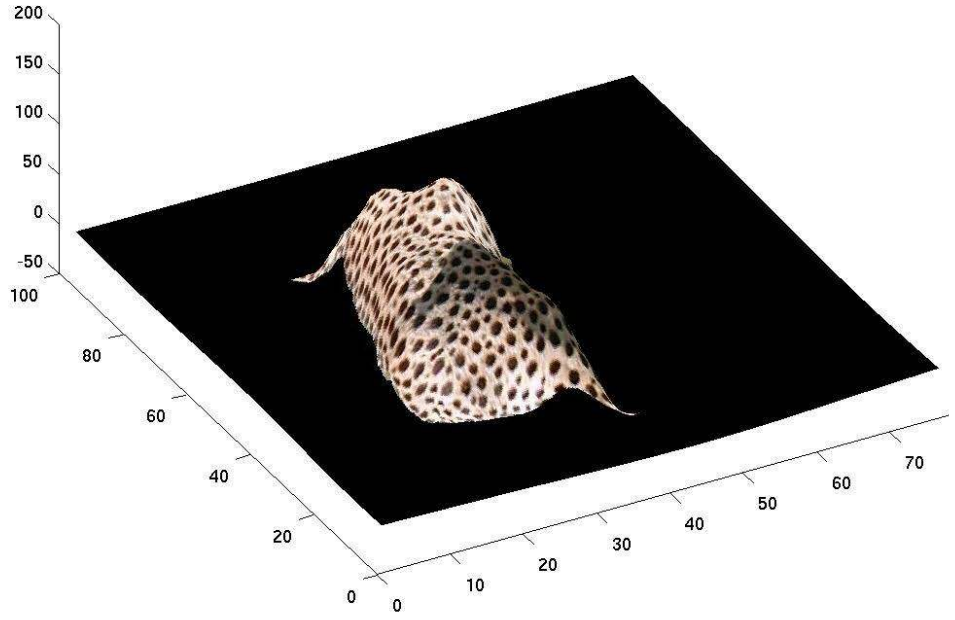
$\sigma_\tau \rightarrow$ major axis / minor axis



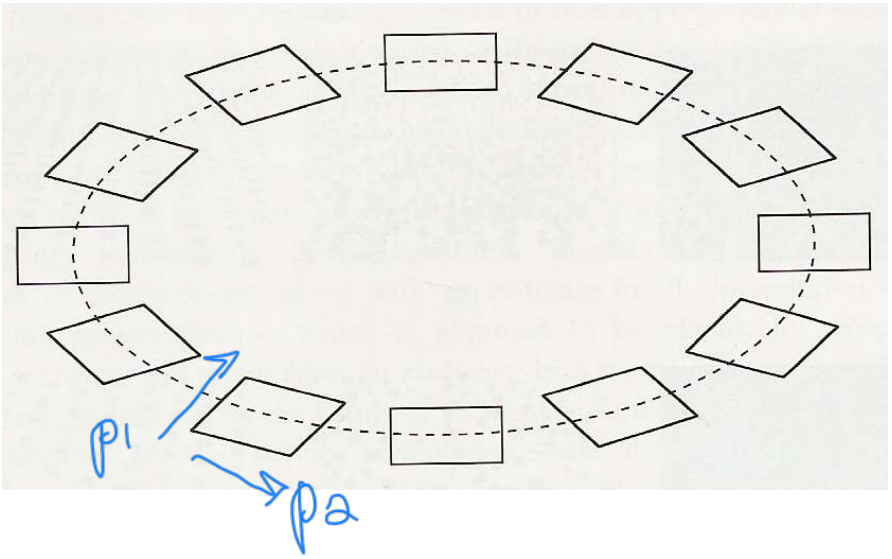
- **Tilt:** Derived from the image direction in which the surface element undergoes maximum compression.
- **Slant:** Derived from the extent of this compression.

Cheetah





Perpendicular Lines



Orthographic projections of squares that are rotated with respect to each other in a plane inclined at ω=60° to the image plane.

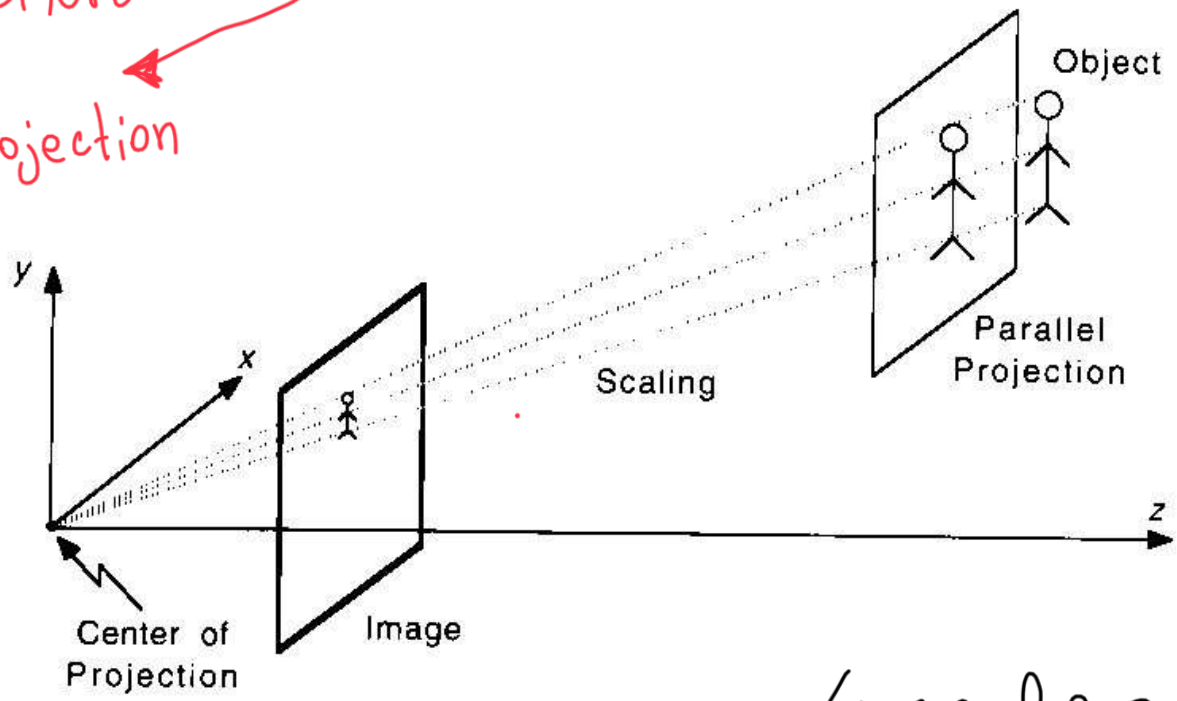
$$\frac{\|\mathbf{p}_1/l_1 \times \mathbf{p}_2/l_2\|}{2} = \frac{\cos(\omega)}{1+\cos^2(\omega)}$$

$\|p_1/l_1\|^2 + \|p_2/l_2\|^2$

Parapespective Projection

perspective projection

orthographic projection



(was good for close to optical axis)

Generalization of the orthographic projection:

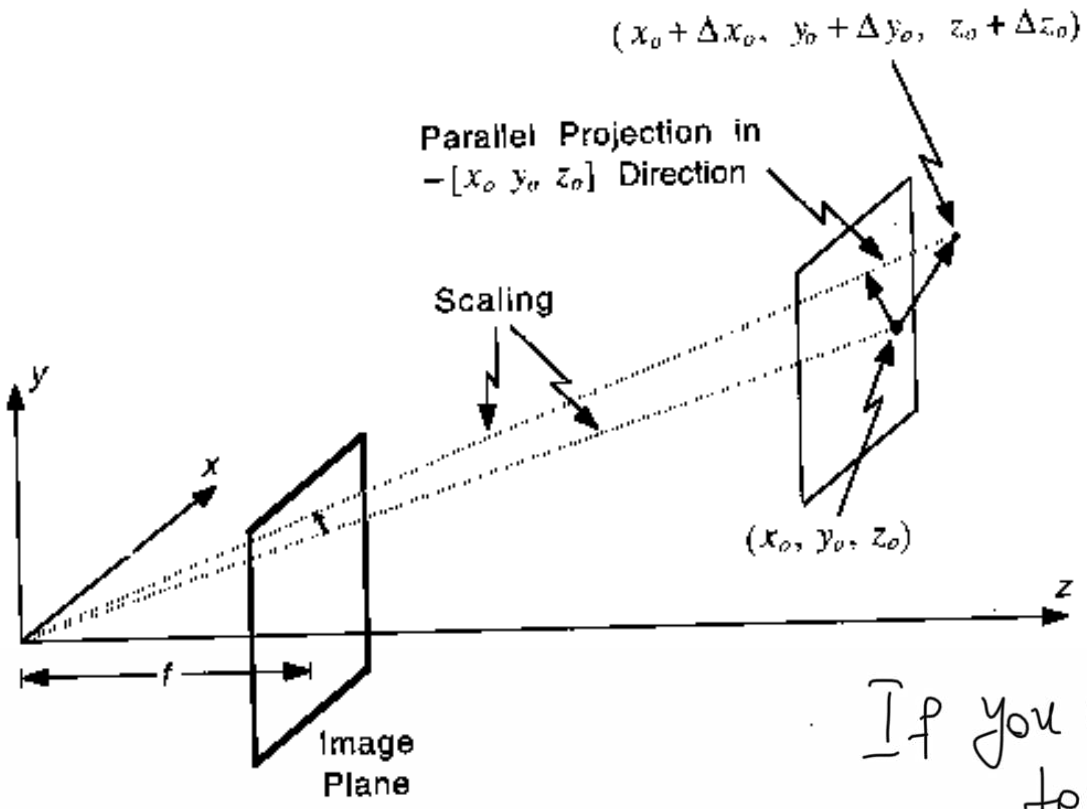
- Object dimensions small wrt distance to the center of projection.

→ Parallel projection followed by scaling

(far away from optical axis)

(good for cameras with long focal length acquired with)

Parapespective Projection



If you compute this for several texture elements

- For planar texels:

Unknown surface normal.

True Area.

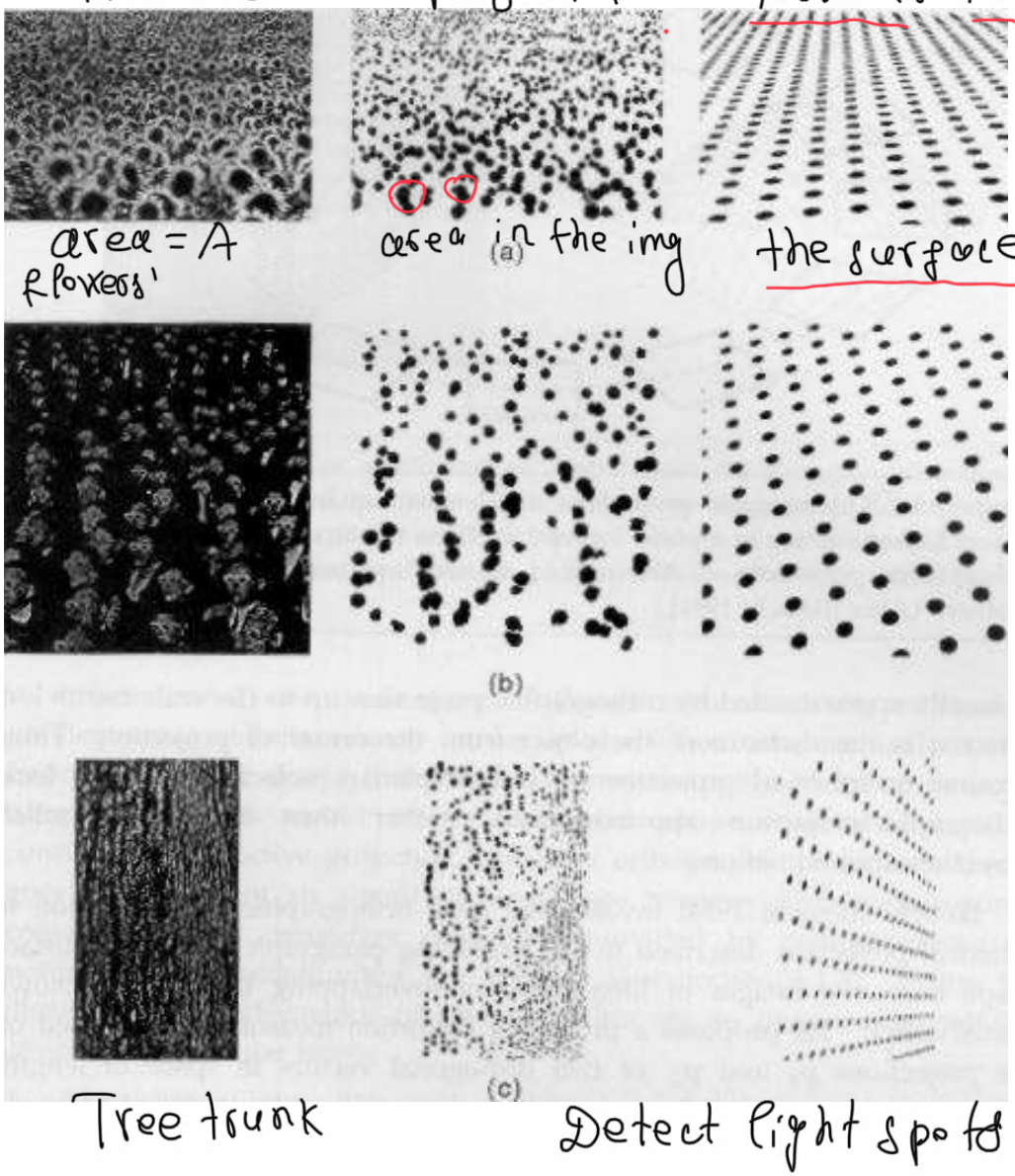
Projected Area. → $$A' = -\frac{f^2}{z_0^3}\,\mathbf{n} \bullet [x_0\, y_0\, z_0] A$$

texels appear smaller as their distance from cam ↑

↑ depth

large when surface faces camera small when surface angled away from cam

Parapespective Projection



You can write eq'ns that connect several eq'ns that connect the area in the world to observed area to the normal

Texels:

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

compute normals with the fact of over-constrained task

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

Bottom line: If you can detect enough texture elements, then you can estimate shape of surface

Texture Gradient

Back, things get compressed
As you go further and further

Textures on the front != textures on the back

So far, we talked about structural texture

Statistical Shape Recovery

textures

Mesure 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.

Density you measure at which Image coordinates.

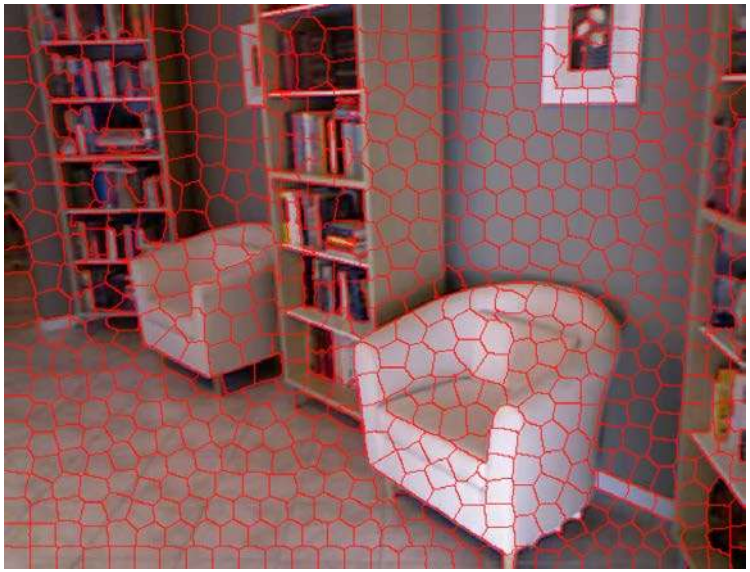
Function of density.

normal

connect densities observed at different parts of images ↔ normal to the plane

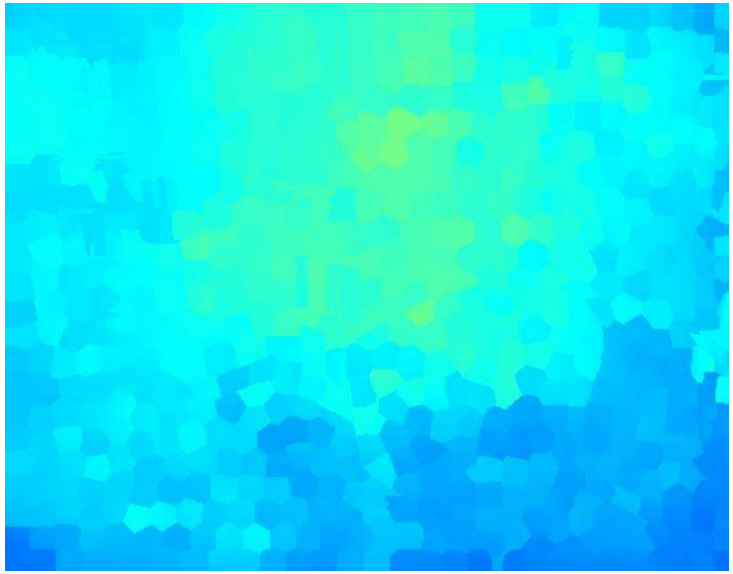
Machine Learning

Input Image



Superpixels

Previously, we were assuming to have circles, squares, some textural elements, some textural densities over whole surface with a single normal



Input: Single Image

Output: Surface Normals

even for non-homogeneous texture

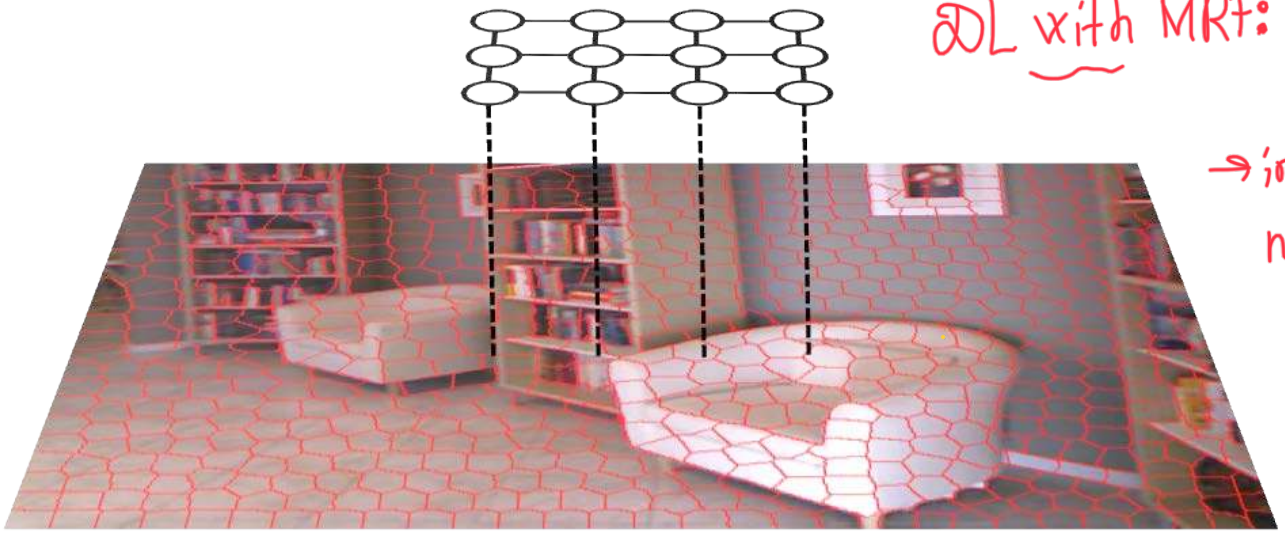
Because we tackled superpixel scope.
If we use ML, results will be noisy

Train a regressor to predict depth —> Noisy predictions

Markov Random Field (MRF)

If you can compute normals => you can integrate and then compute 3D shape

Graph with vertices and edges



DL with MRF: → explicitly consider image as a graph (of superpixels)

→ impose Potts on its neighbouring consistency

Assign values to the nodes to minimize

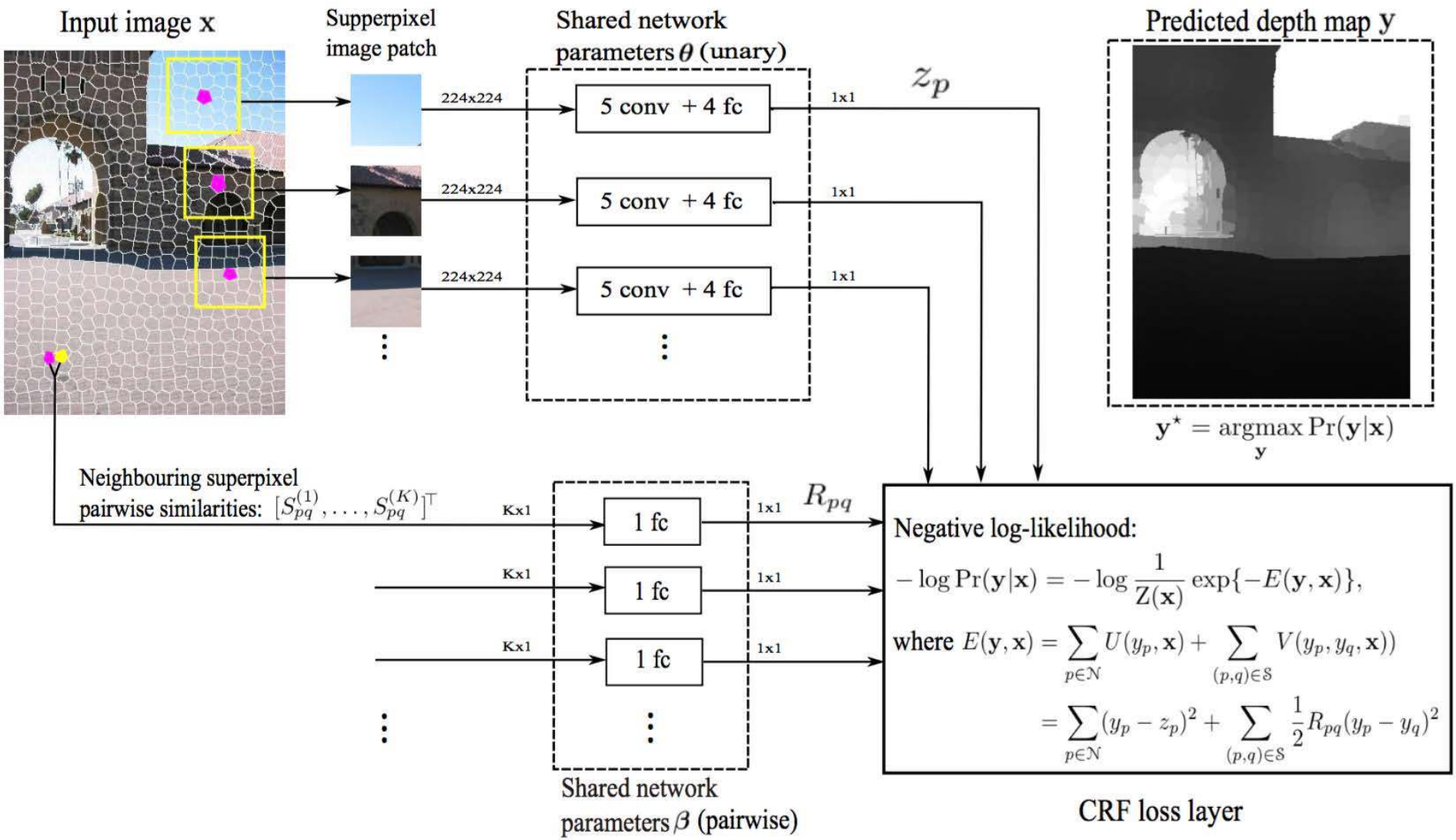
$$E(Y) = \sum_{i} \varphi(y_i) + \sum_{(i,j)} \psi(y_i, y_j)$$

unary pairwise

Why: Adding some spatial consistency Potts enforces connectivity and smoothness

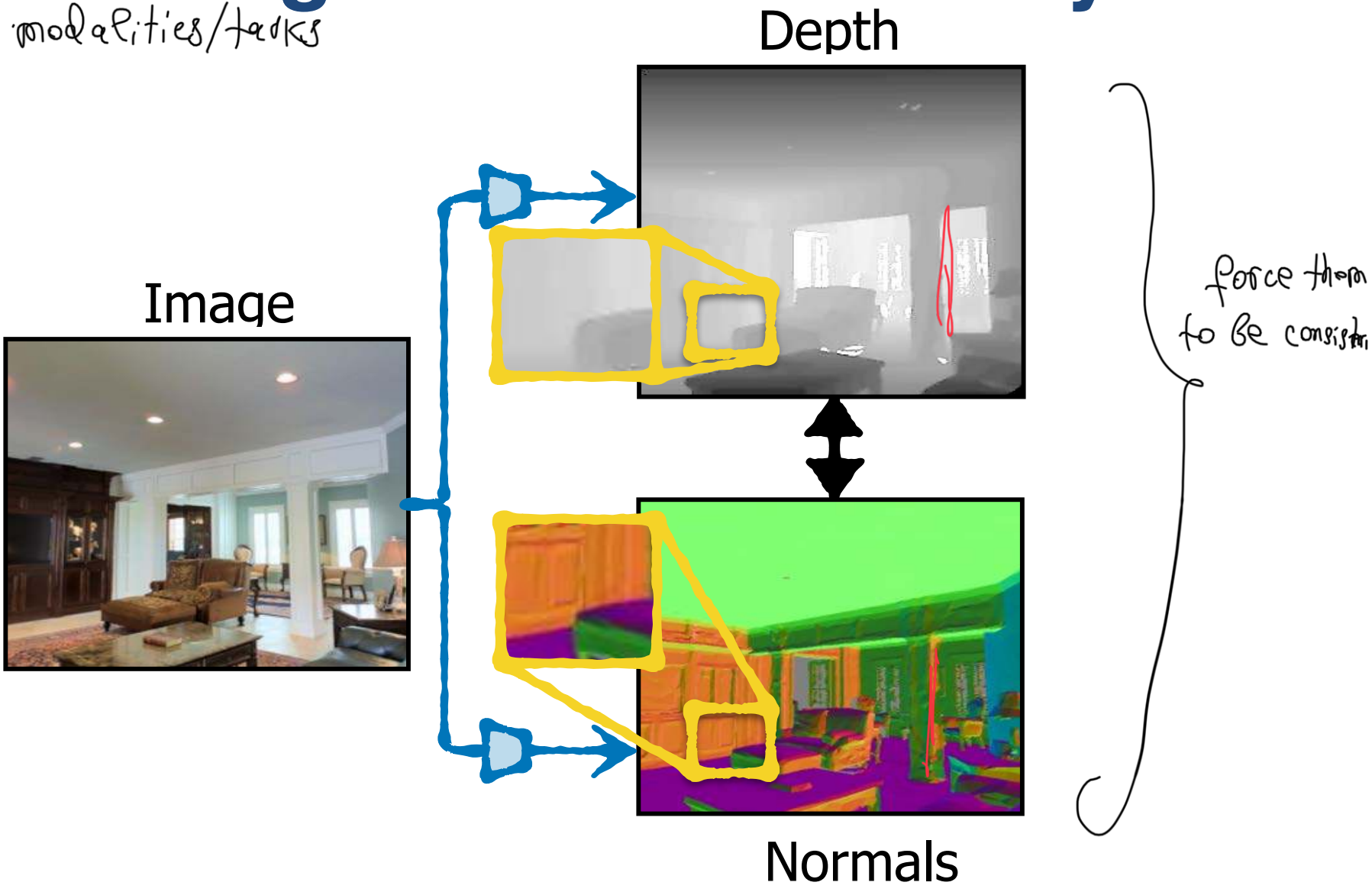
—> Enforces consistency

Deep Learning with MRF



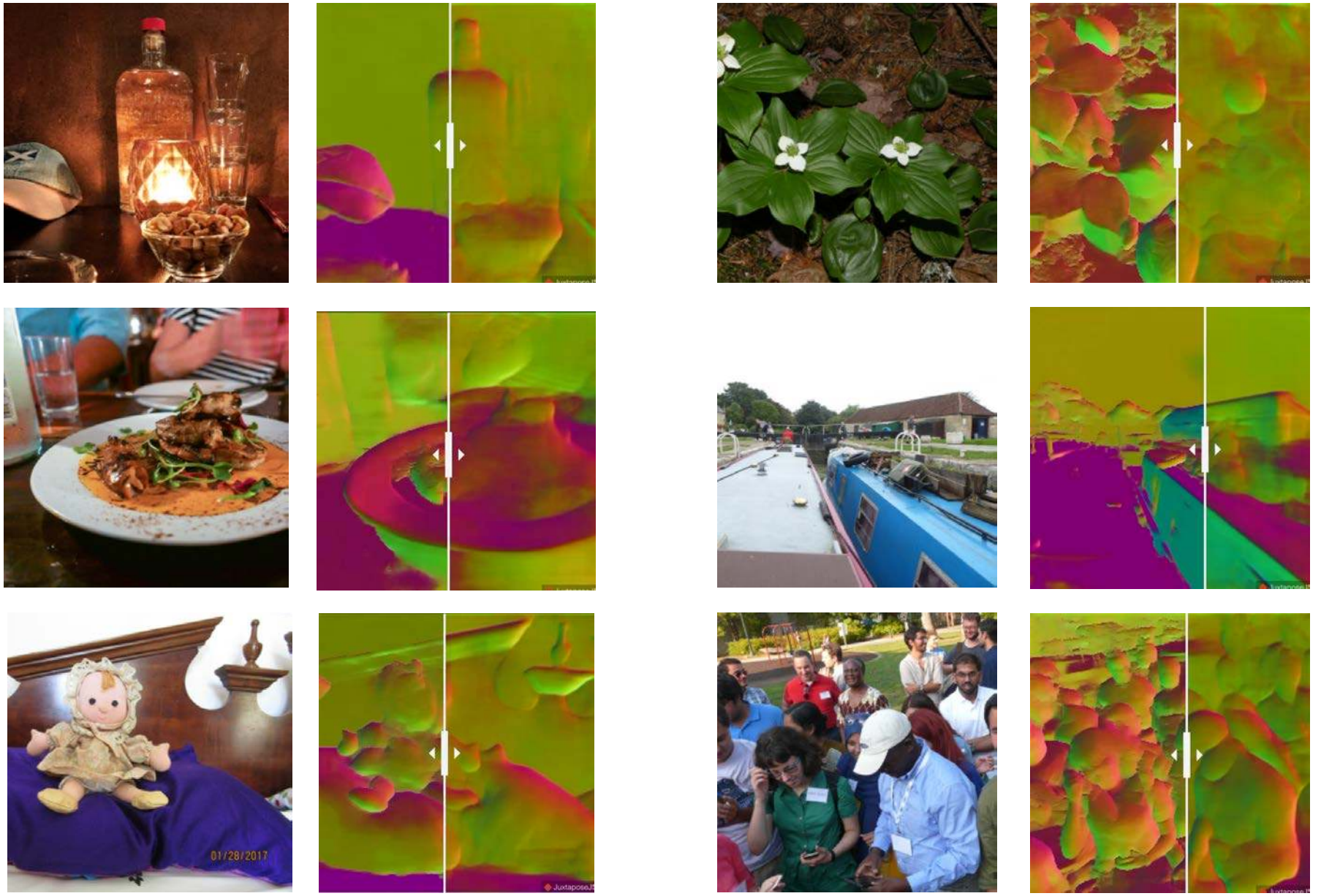
Enforcing Task Consistency

Across modalities/tasks



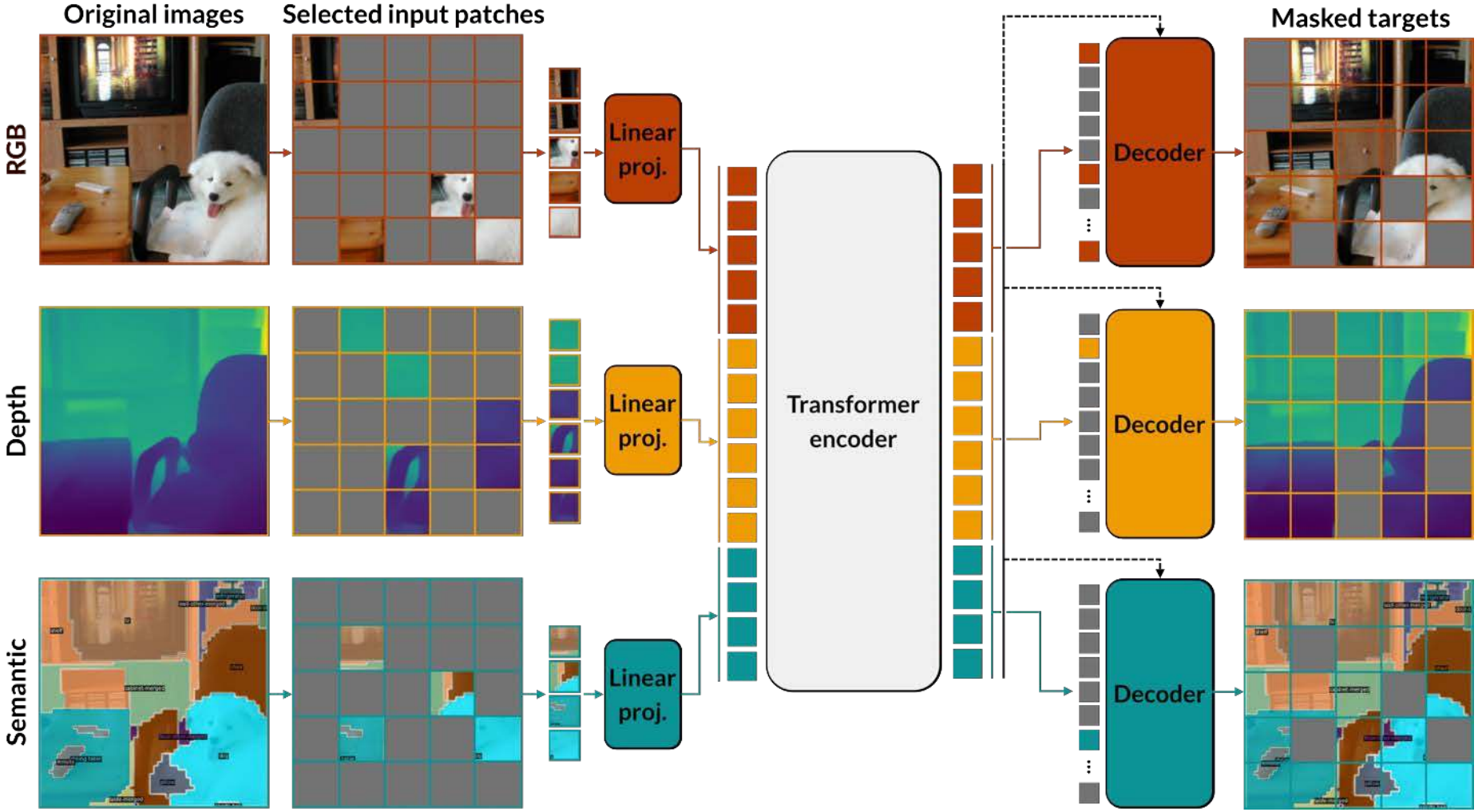
- A network can be trained to predict multiple things.
- Forcing consistency across tasks increases robustness.

A Very Diverse Training Database Helps



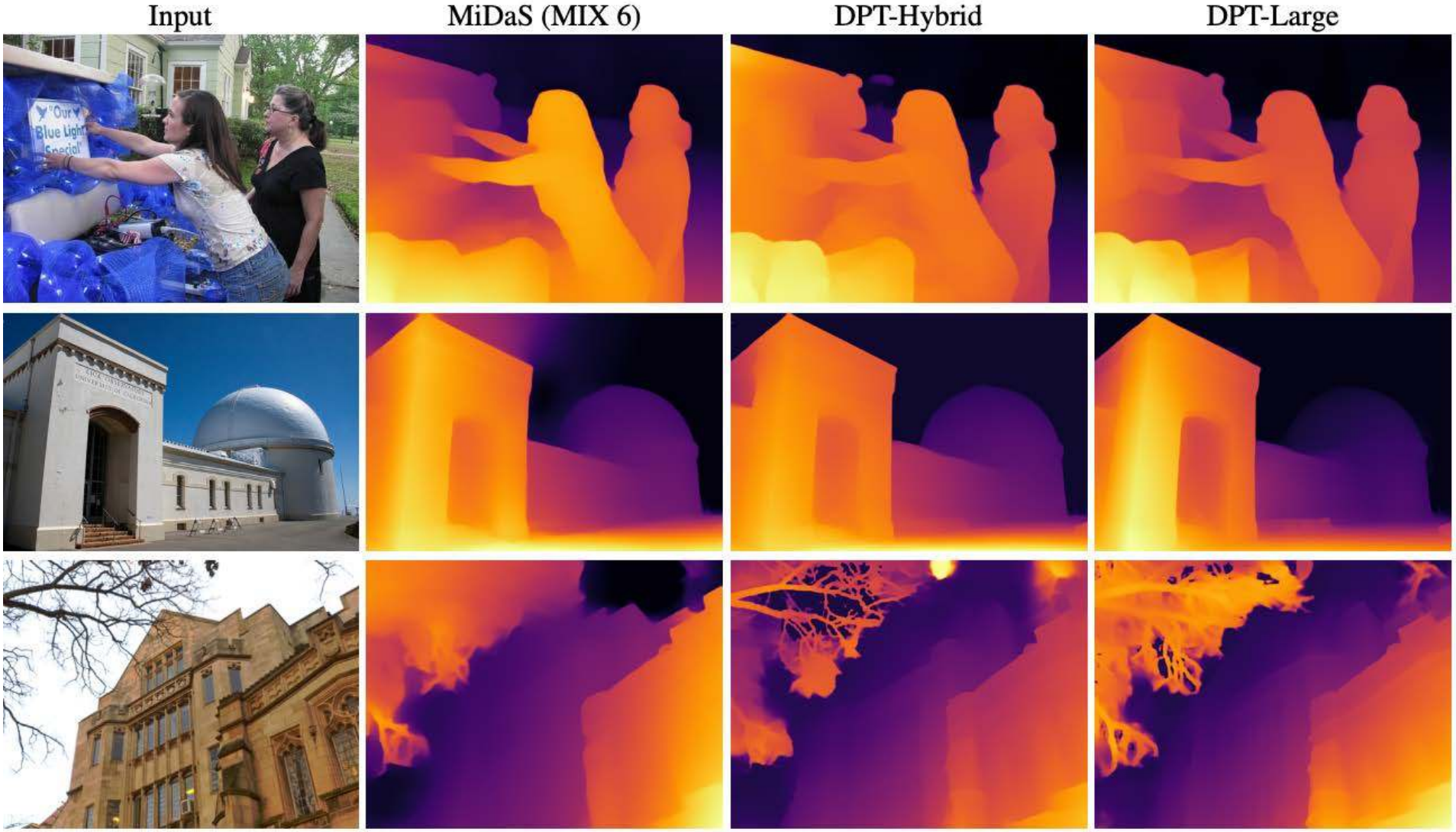
Eftekhar et al. , ICCV'21 vs Chen et al. , CVPR'20

.. and so does a Transformer Architecture



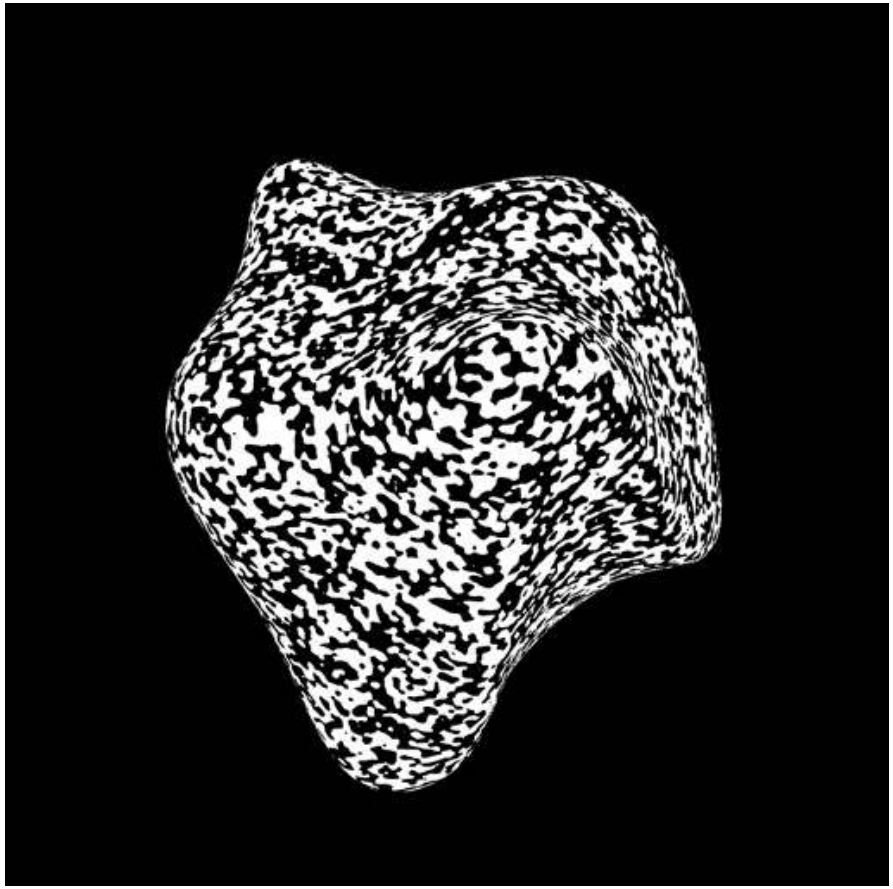
Ranftl et al, ICCV'21

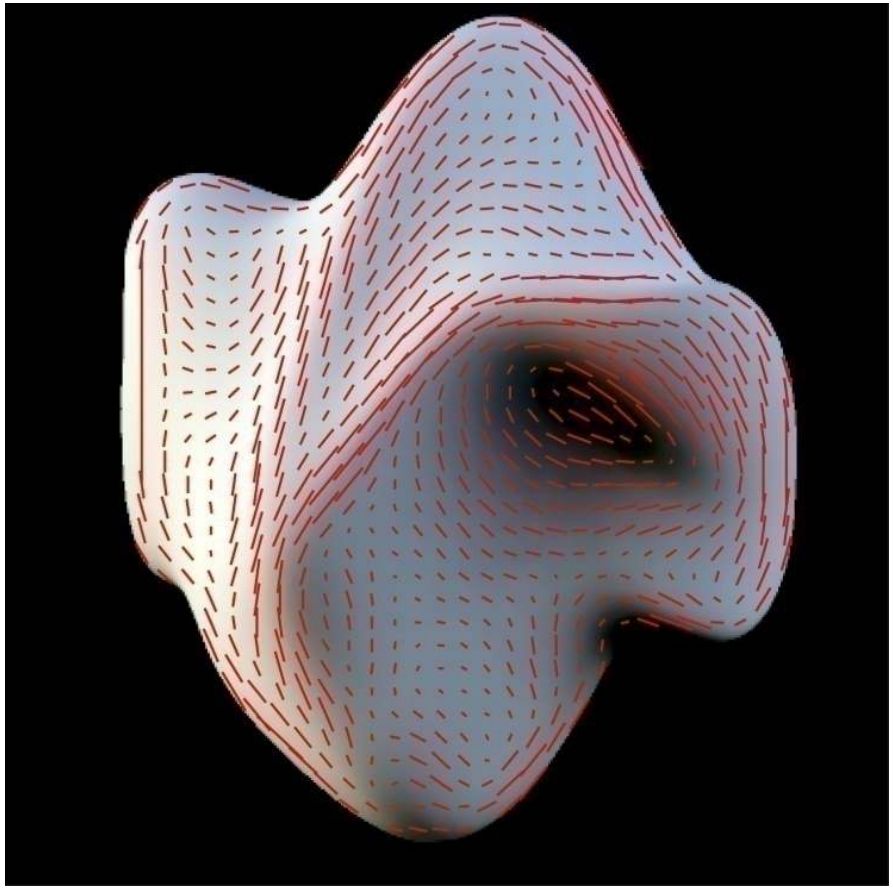
Using Transformers



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

Optional: Illusory Shape Distorsion





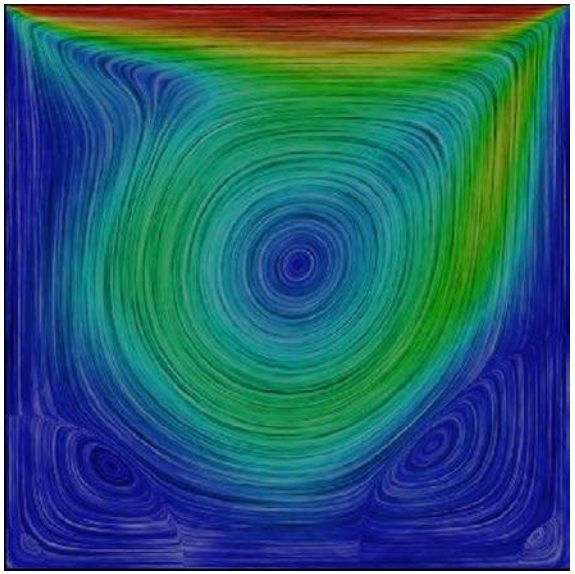
People seem to be sensitive to orientation fields in the cases of both texture and shading.

Flemming et al. PNAS'10

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.

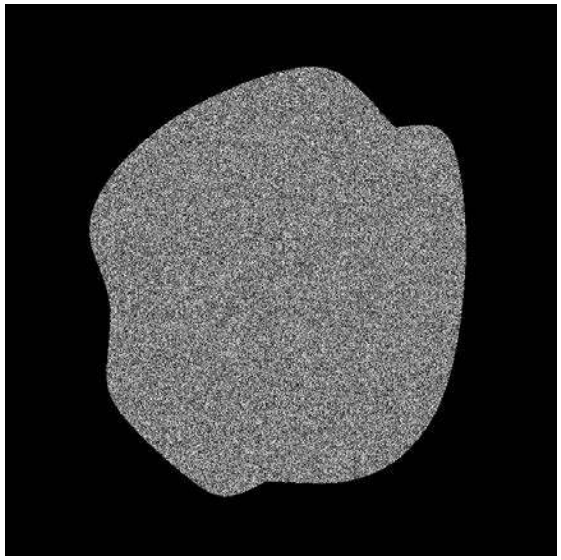
Test: Use a technique known as Line Integral Convolution to smear the texture along specific orientations and scale appropriately.

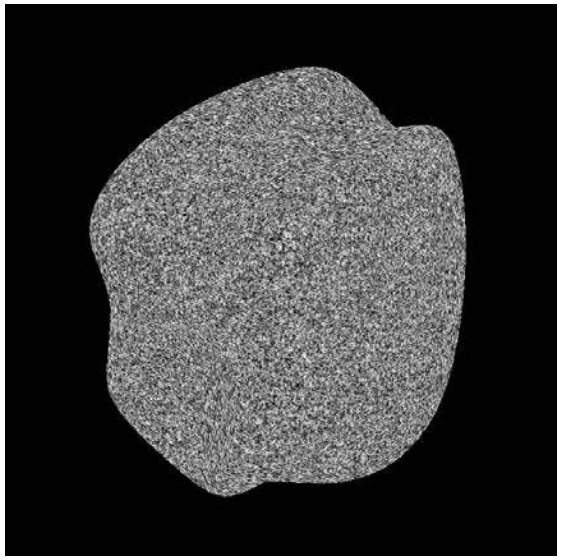


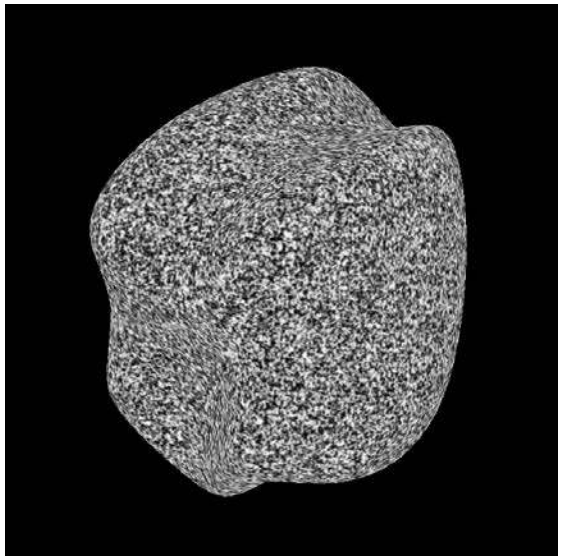
Flemming et al. PNAS'10

Optional: Scaling and Smearing

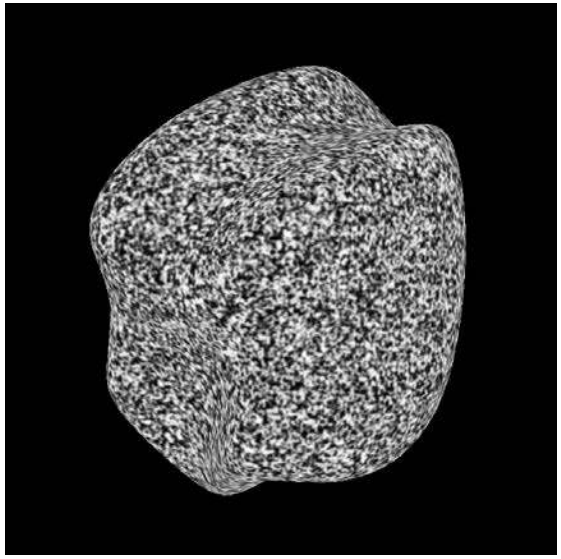
Scaling:

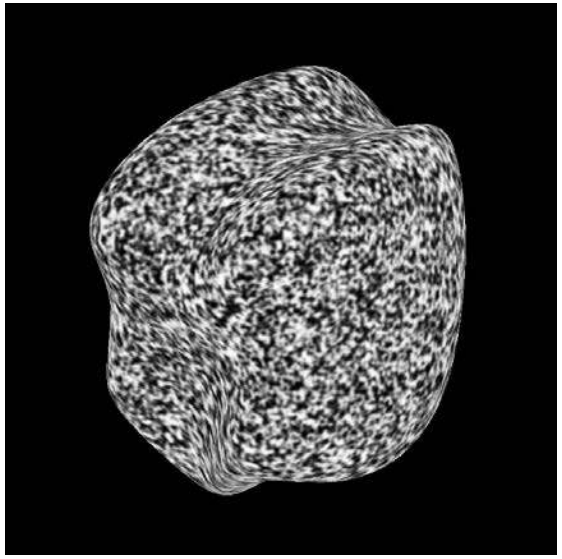


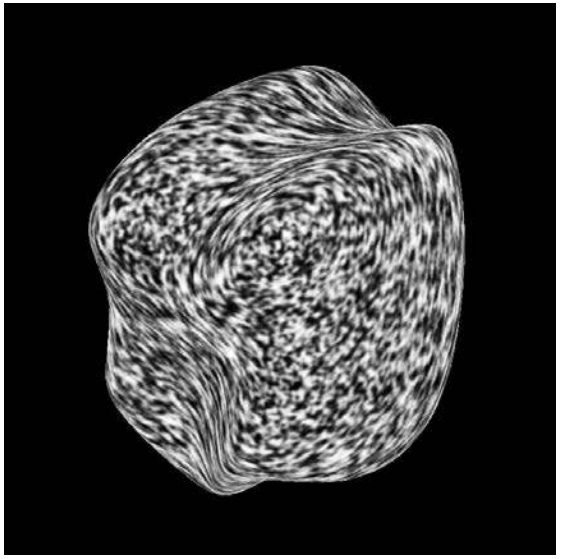




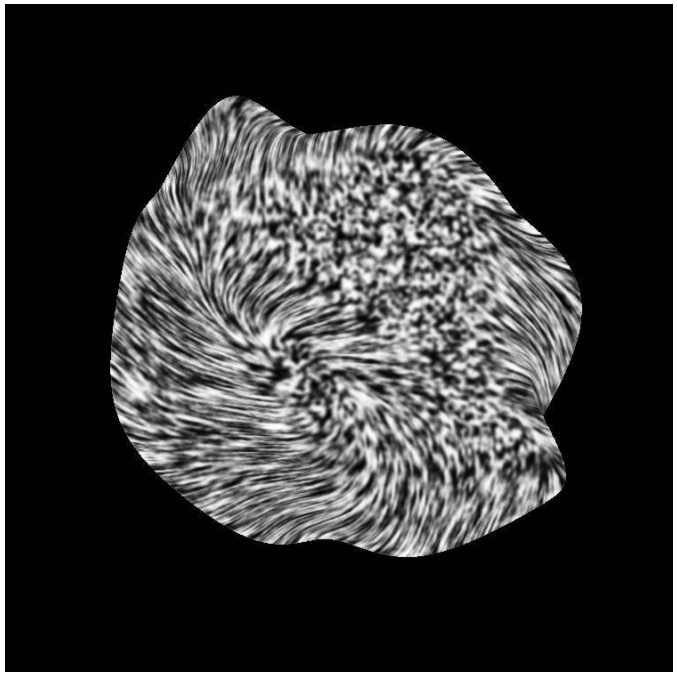
Smearing:

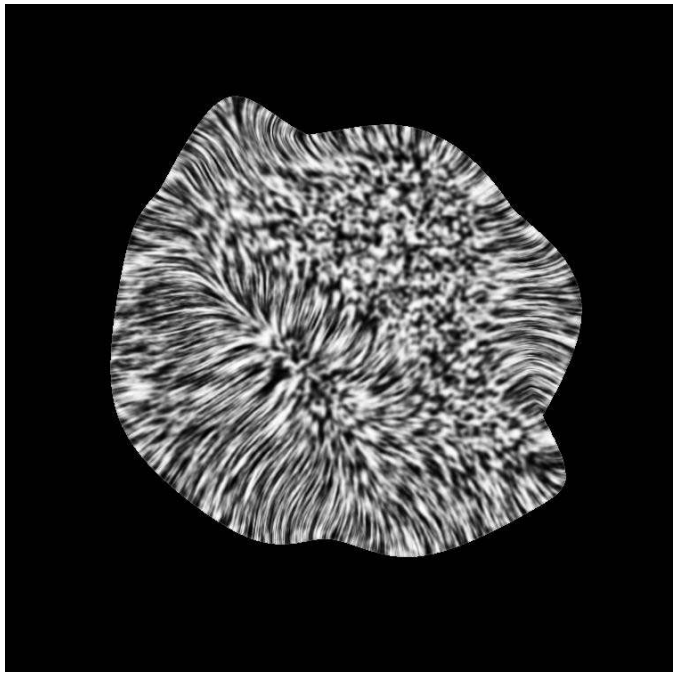






Optional: Inconsistent Stimulus





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. limited set of cases
- Deep learning can be used to weaken them.

trained on specific settings