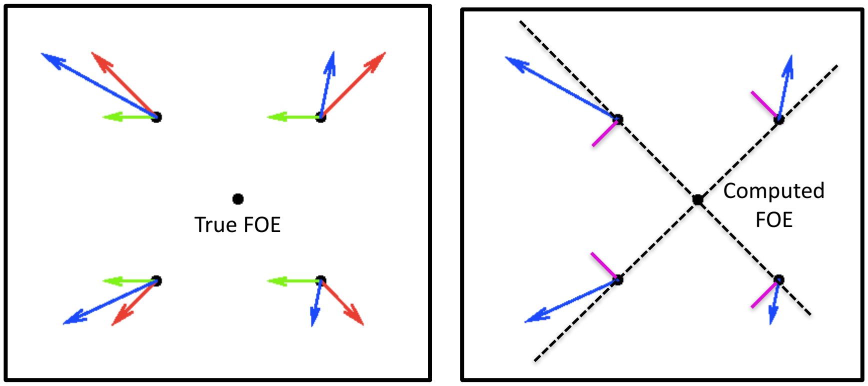
Recovering the observer's rotation



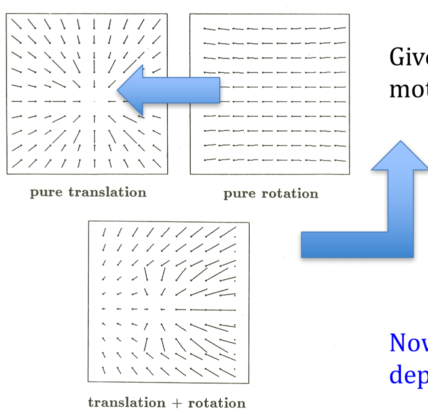
Velocity component due to observer's translation
Velocity component due to observer's rotation
Final velocity at each location

Computed FOE → *translational field lines*

Velocity perpendicular to field lines must be due to observer's rotation!

Find (R_x, R_y, R_z) that ***best explains*** the motion perpendicular to the field lines

Finally, recovering 3D layout



Given (R_x, R_y, R_z), compute image motions due to rotation…

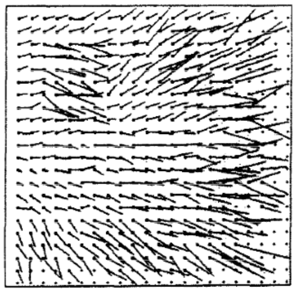
… then subtract motions due to rotation, to obtain the image motions due to the observer's translation

Now, how can we compute the relative depth of surfaces in the scene?

What are we assuming about objects in the scene?

When is this assumption violated?

Detecting moving objects



Noisy velocity field, computed from a scene of rectangular objects at different depths

Locations with large velocity changes in the neighborhood – around object boundaries, where depth changes occur

Directions of velocity differences, with the "best FOE location" and moving objects