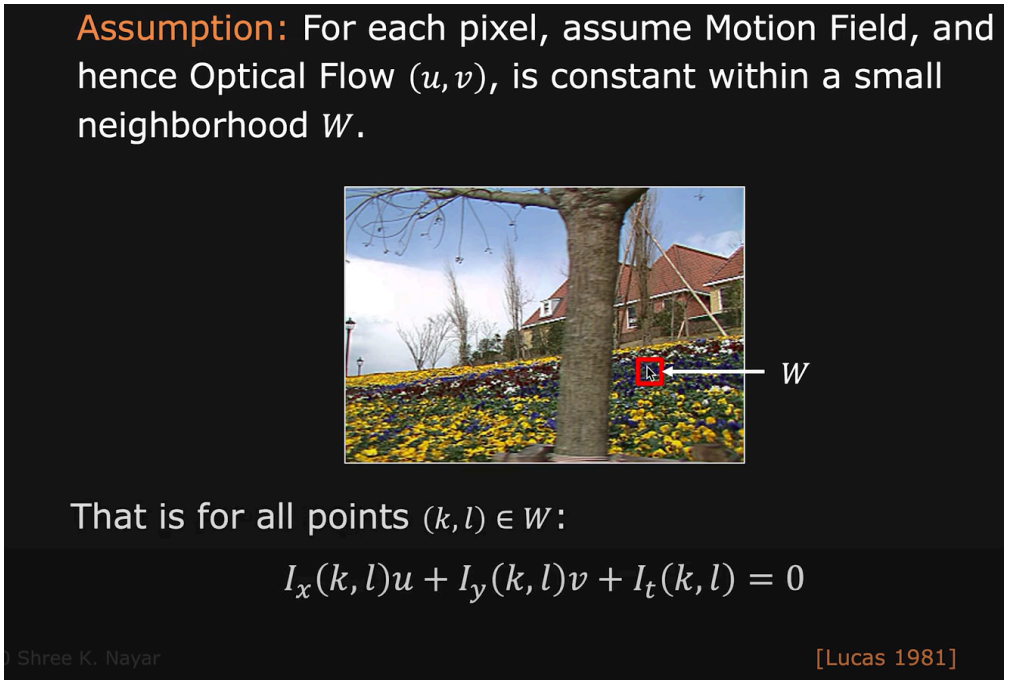
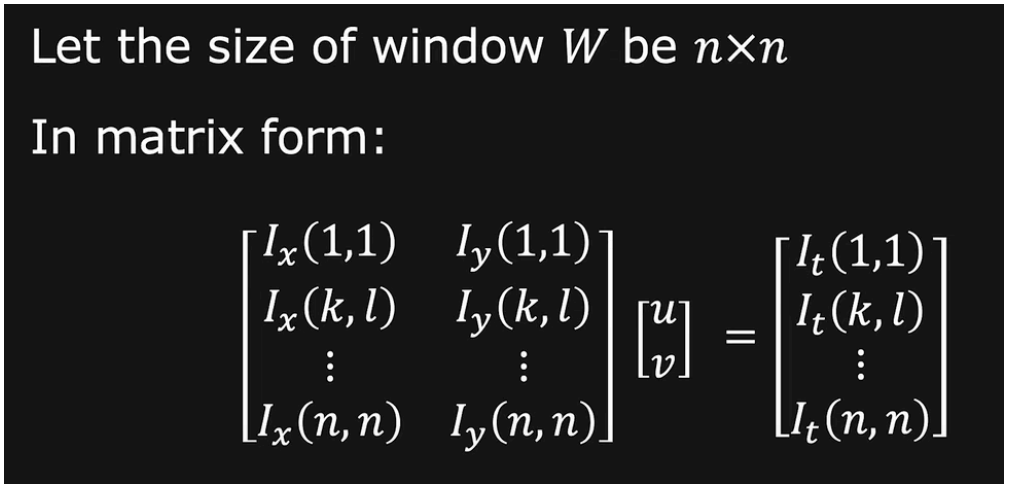
Lucas Kanade

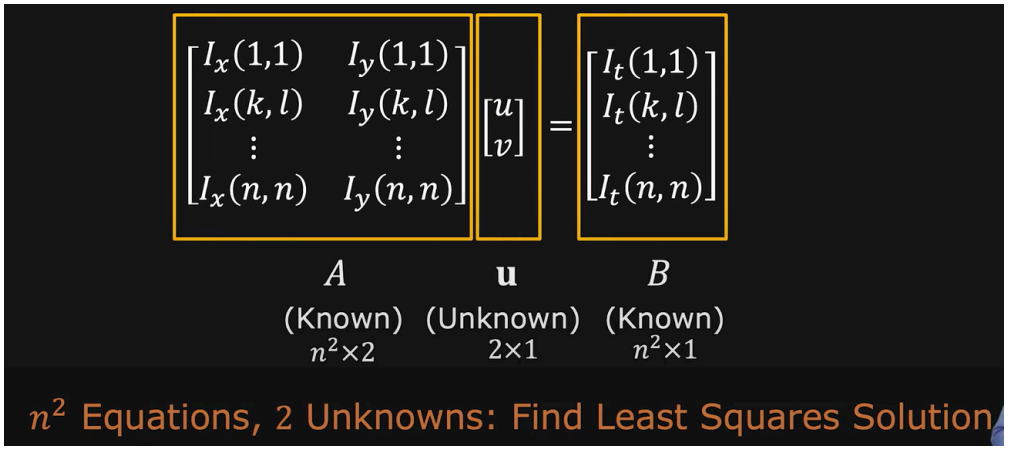
Adds additional constraint that for each pixel within a small local neighborhood W, assume that Motion Field and hence Optical Flow are constant

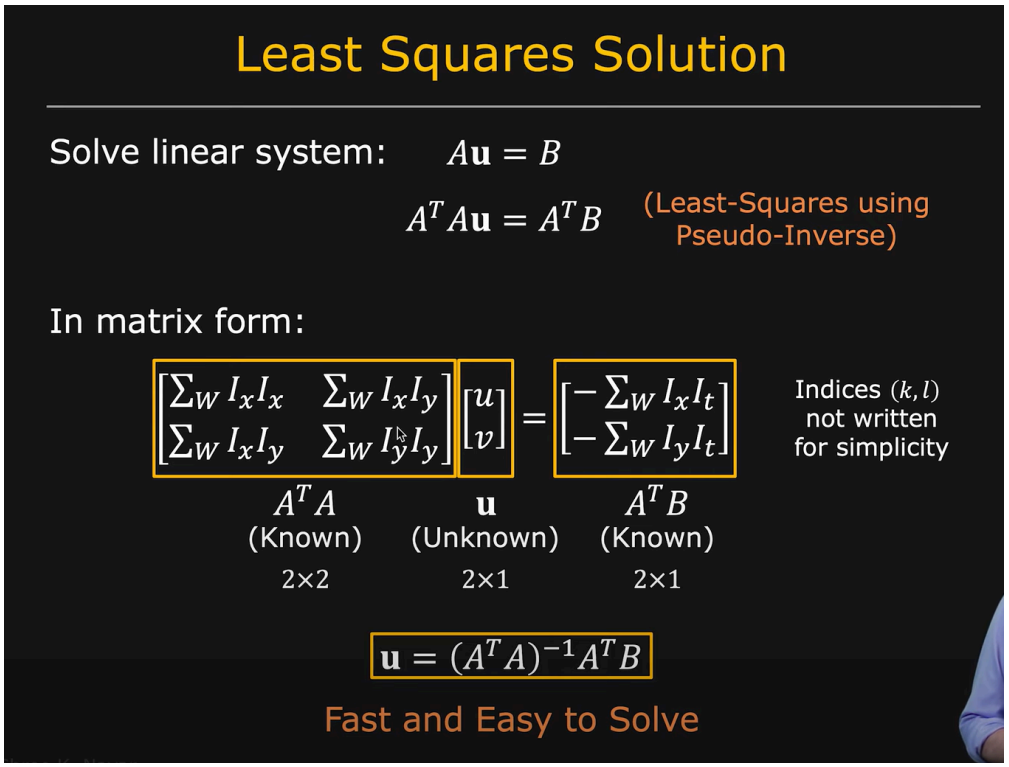


So, now you get a system of equations where the number of equations will be the size of the window.

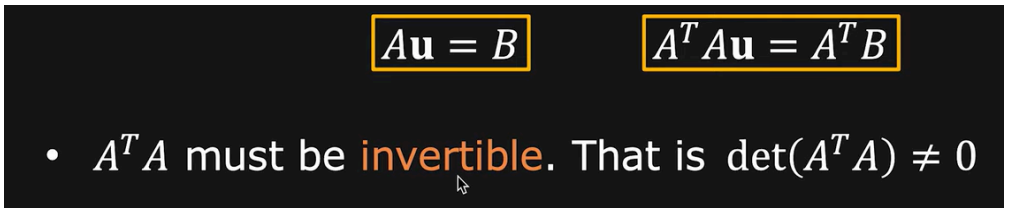


So, why does this work? Why would this matrix on the left be invertible? All of these equations tend not to be linearly dependent on one another. Because if you have a patch that has an interesting texture, then your Ix's and Iy's would most likely be different from pixel to pixel.





When would this not work again? What if AT @ A is not invertible?



More than wanting invertibility, we want A.T @ A to be “well-conditioned”

GPT: **Well-conditioned** means that the solution is *stable* with respect to noise or small changes in the input. A poorly conditioned (but still invertible) matrix can produce huge variations in the output for tiny variations in the input.

Also GPT:
Well-conditionedness is important because it ensures
stability in the solution. In the context of Lucas-Kanade, you are solving a system of linear equations derived from image gradients. If that system's coefficient matrix (often the structure tensor) is **ill-conditioned**, tiny amounts of noise or small changes in the gradients can lead to **large errors** in the computed optical flow. A well-conditioned matrix, on the other hand, keeps the solution more **robust** and **less sensitive** to inevitable noise or approximations in the image data.

- A^TA must be well-conditioned.

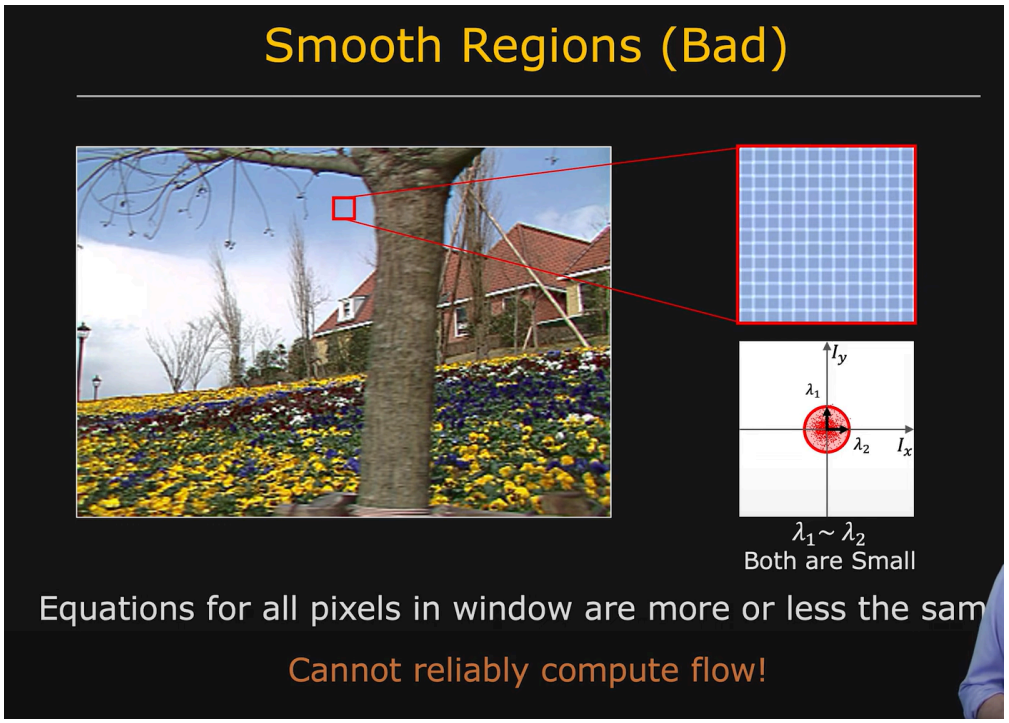
  If λ_1 and λ_2 are eigen values of A^TA, then

  $\lambda_1 > \epsilon$ and $\lambda_2 > \epsilon$

  $\lambda_1 \geq \lambda_2$ but not $\lambda_1 \ggg \lambda_2$

For A.T @ A to be well conditioned, the eigen values must be significant enough. But just that one eigen value cannot be SIGNIFICANTLY larger than the other one.

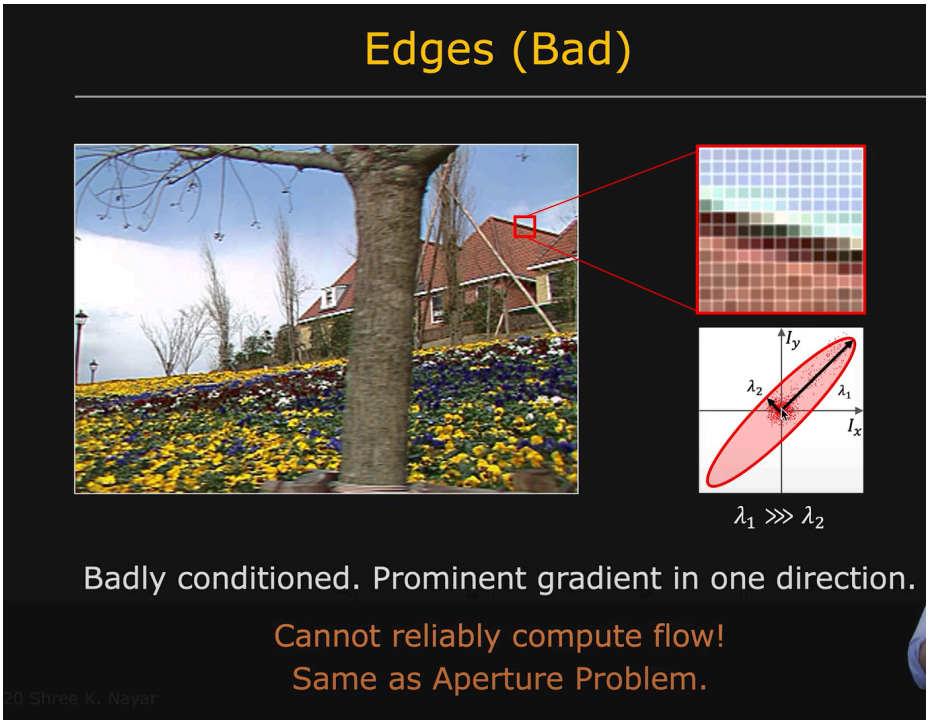
How non well-conditioned matrices manifest IRL



Note that A.T @ T is the covariance matrix and it's eigen vectors will show the direction where the data varies the most. If you have a smooth region then the Ix and Iy values are very small (these are the spatial gradients) then they will form a cloud round the origin. To find the eigen vectors of A.T @ T, fit an ellipse to these points. Then the semi major and major axis are the eigenvectors (of the covariance matrix A.T @ A). The covariance matrix also has very small eigen values.

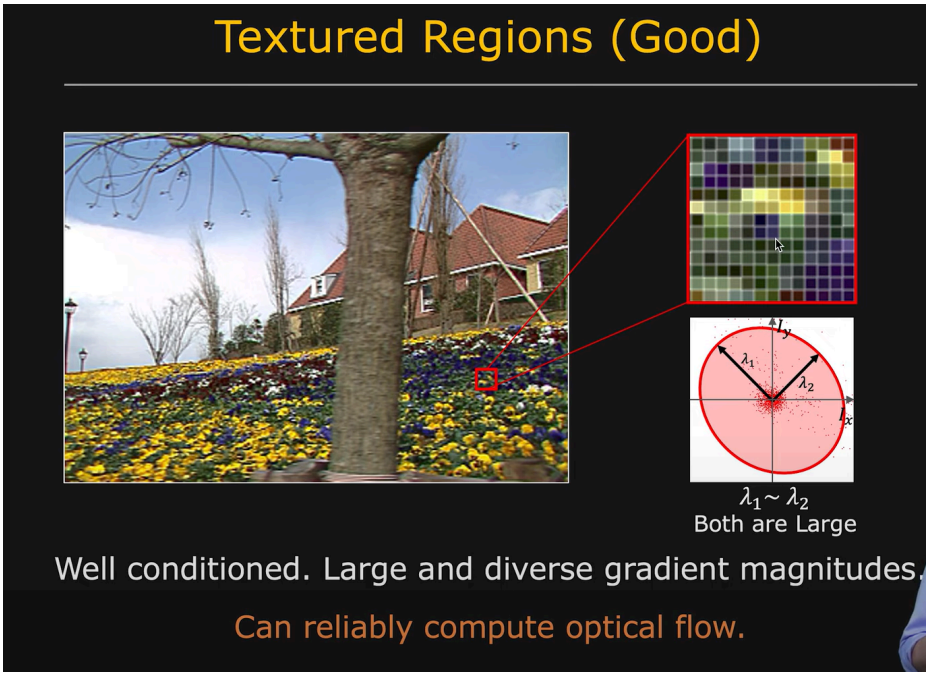
So optical flow cannot be computed reliably.

Another example of a bad region:



Edges are also bad because you have strong gradients (of pairs (Ix, Iy)) in one direction, so one of the eigen values is much larger than the other one.

How do well conditioned matrices manifest IRL?



Good changes in brightness in both directions, so pairs of spatial gradient pairs (Ix, Iy) are well scattered. So that the eigen values are large but not large w.r.t each other. Therefore the matrix A.T @ T is well conditioned so that the optical flow computed is reliable.