Lecture 5.3
The perspective camera model revisited

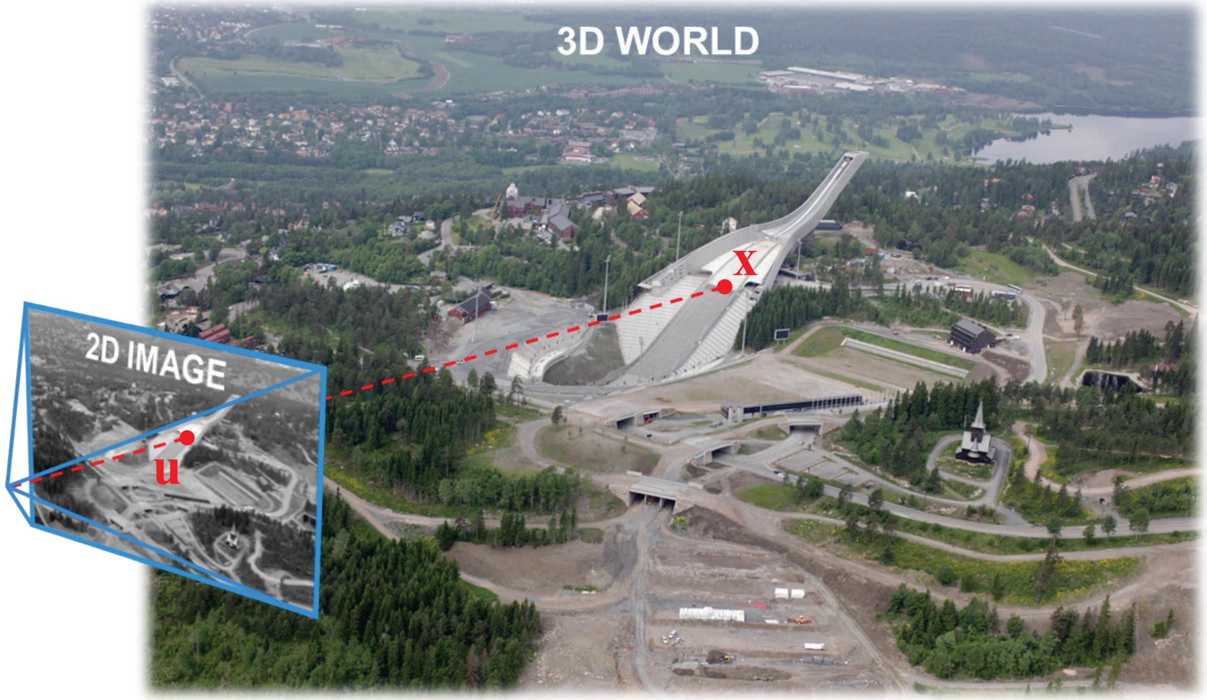
Thomas Opsahl

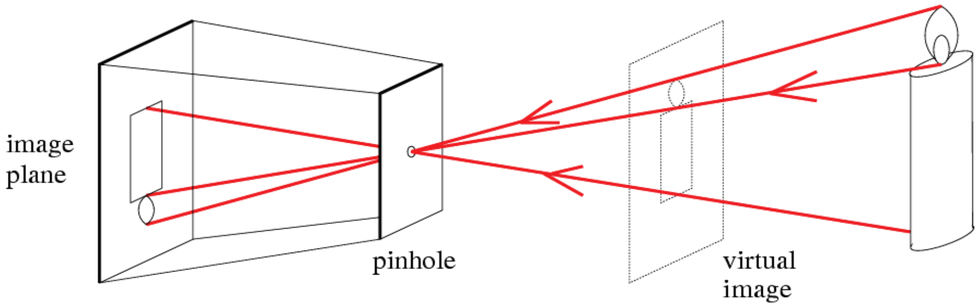
The perspective camera model

A mathematical model that describes the viewing geometry of pinhole cameras

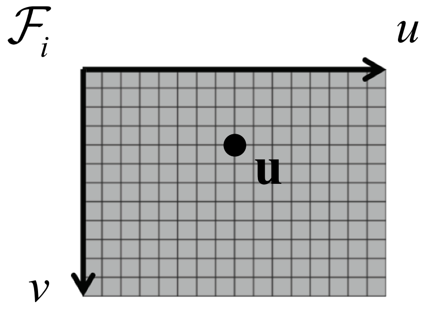
It describes how the perspective projection maps 3D points in the world to 2D points in the image

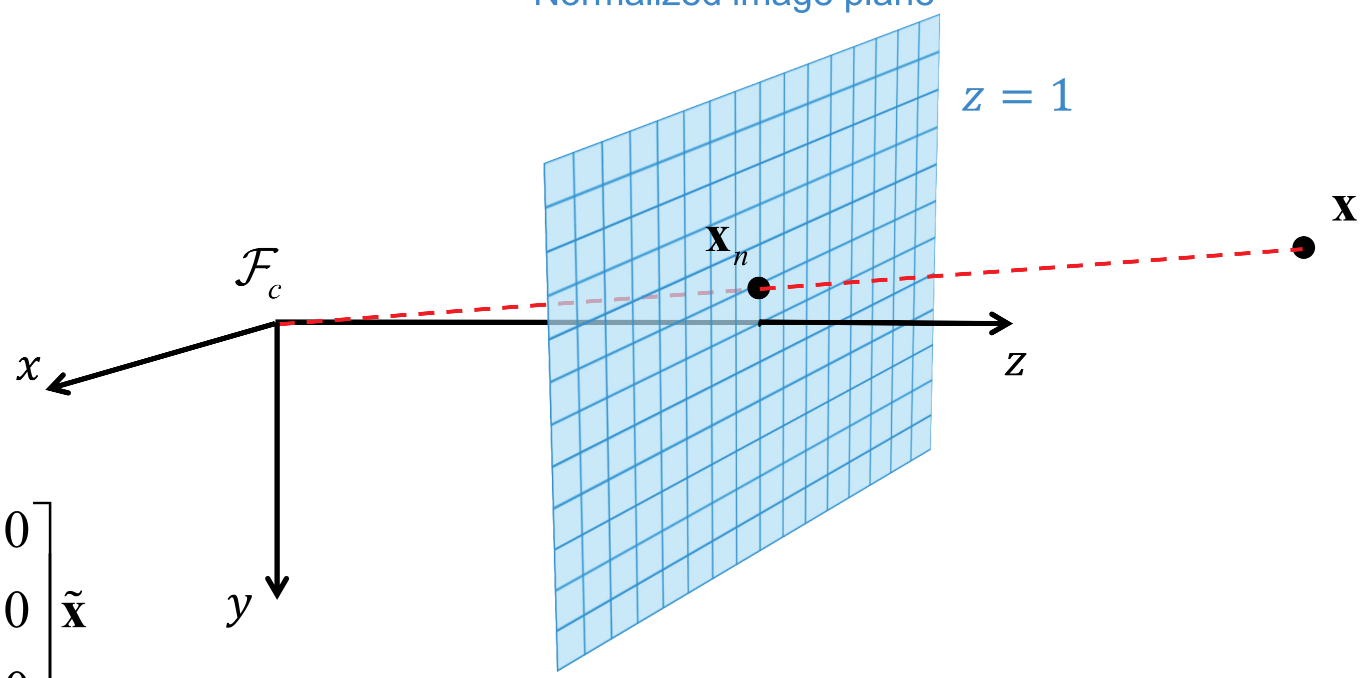
Combined with a distortion model, the perspective camera model can describe the viewing geometry of most cameras





The perspective camera model

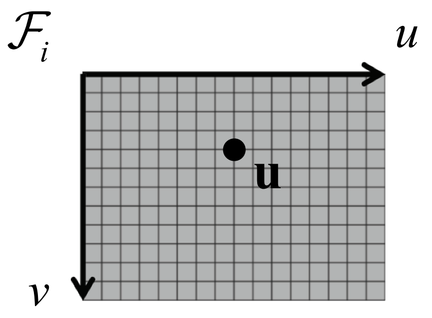




$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \tilde{\mathbf{x}}$$

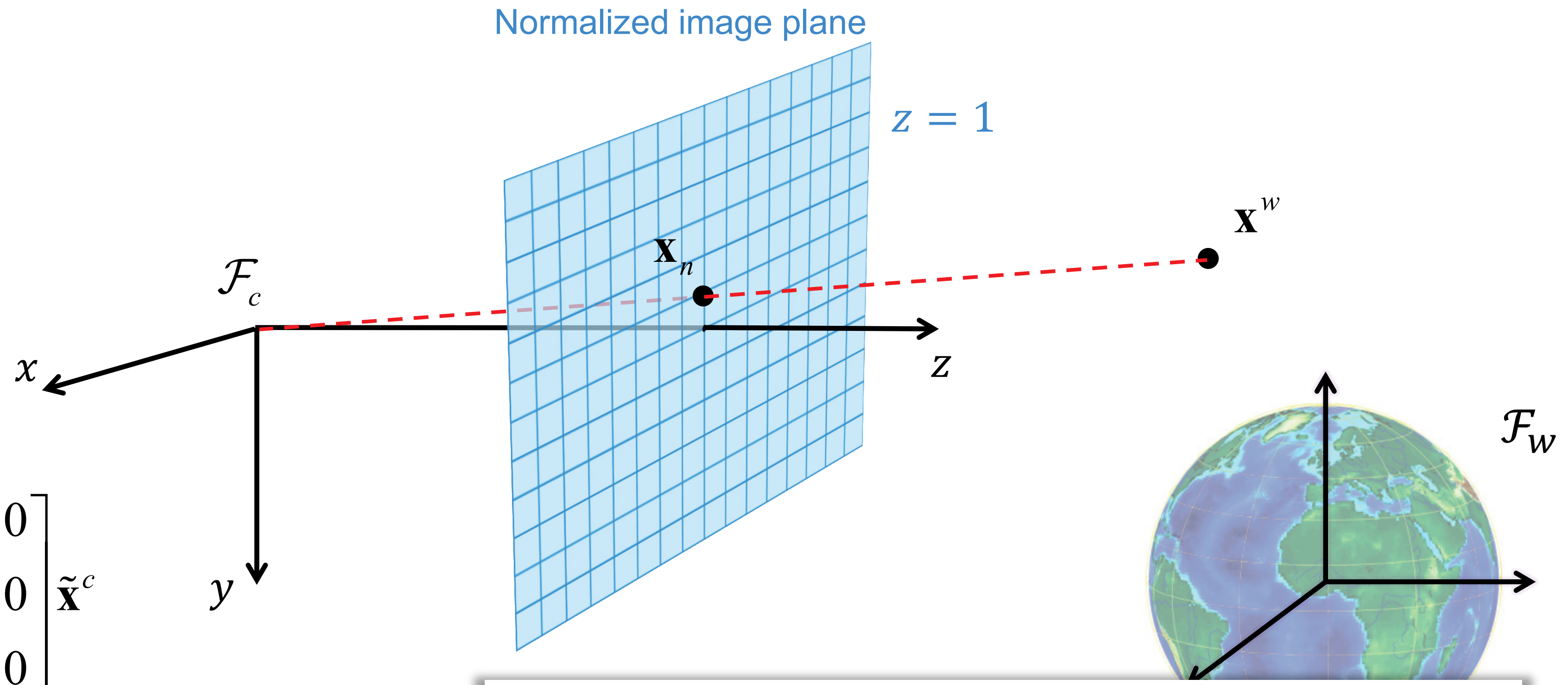
$\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{x}}_n$	$\tilde{\mathbf{x}}_n \leftarrow \tilde{\mathbf{x}}$

The perspective camera model



$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \tilde{\mathbf{x}}^c$$

$\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{x}}_n$ $\quad$ $\tilde{\mathbf{x}}_n \leftarrow \tilde{\mathbf{x}}^c$



In the current description, the world point is given relative to the camera frame $\mathcal{F}_c$

What changes if the world point is given relative to another coordinate frame $\mathcal{F}_w$?

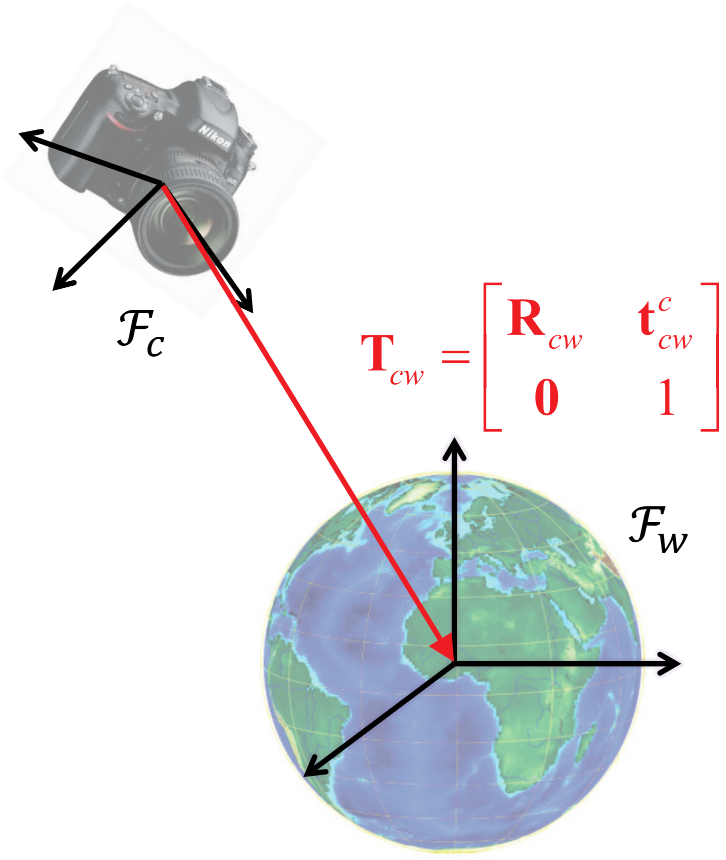
The perspective camera model

- The pose of the world frame relative to the camera frame, denoted by $\mathbf{T}_{cw}$, is also a point transformation from $\mathcal{F}_w$ to $\mathcal{F}_c$

- General perspective camera model

$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \underbrace{\begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw}^c \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}_{cw}} \tilde{\mathbf{x}}^w$$

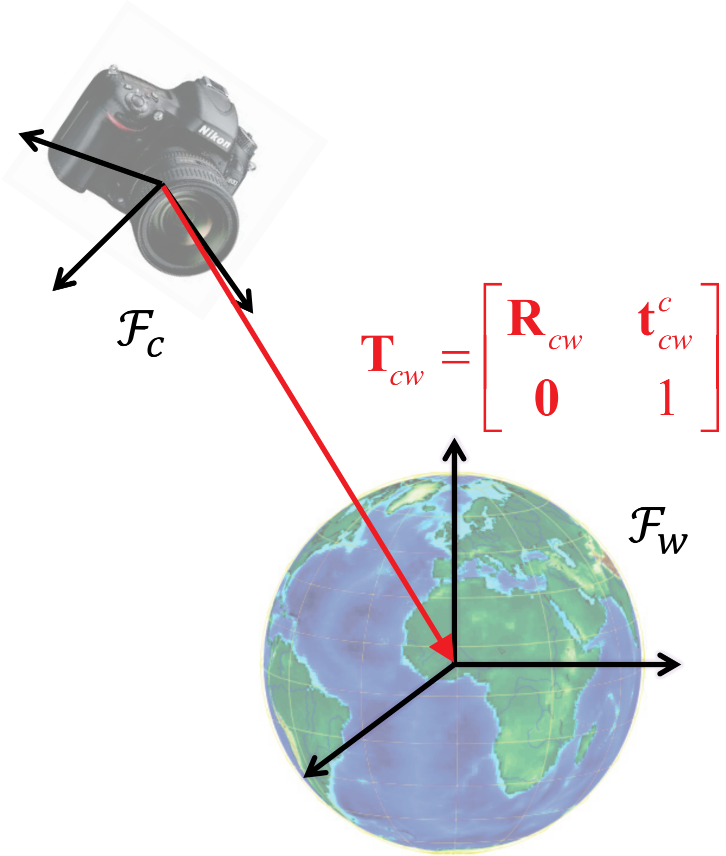
$\tilde{\mathbf{u}} \leftarrow \tilde{\mathbf{x}}_n$	$\tilde{\mathbf{x}}_n \leftarrow \tilde{\mathbf{x}}^c$	$\tilde{\mathbf{x}}^c \leftarrow \tilde{\mathbf{x}}^w$



The perspective camera model

- By multiplying $\mathbf{\Pi}_0$ with $\mathbf{T}_{cw}$ we get a very compact expression that is commonly used to represent the perspective camera model

$$\tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R}_{cw} & \mathbf{t}_{cw}^{c}\end{bmatrix}\tilde{\mathbf{x}}^{w}$$



The perspective camera model

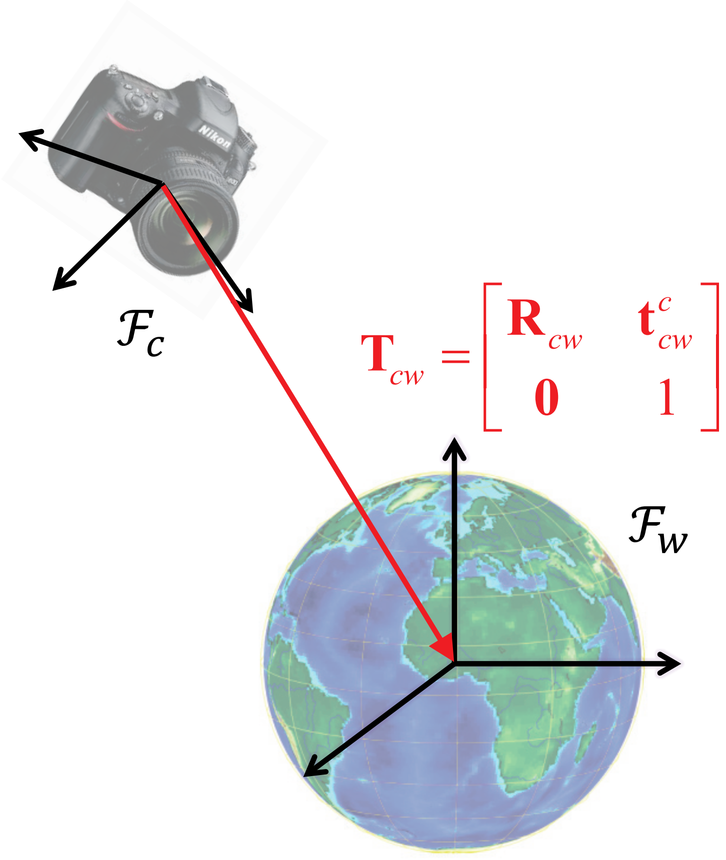
- By multiplying $\mathbf{\Pi}_0$ with $\mathbf{T}_{cw}$ we get a very compact expression that is commonly used to represent the perspective camera model

$$\tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R}_{cw} & \mathbf{t}_{cw}^{c}\end{bmatrix}\tilde{\mathbf{x}}^{w}$$

- We refer to $\mathbf{K}$ as the **intrinsic** part and $[\mathbf{R}_{\mathrm{cw}} \quad \mathbf{t}_{\mathrm{cw}}^{\mathrm{c}}]$ as the **extrinsic** part of the perspective camera model

- The matrix $\mathbf{K}[\mathbf{R}_{\mathrm{cw}} \quad \mathbf{t}_{\mathrm{cw}}^{\mathrm{c}}]$ is often denoted by $\mathbf{P}$ and referred to as camera's projection matrix

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R}_{cw} & \mathbf{t}_{cw}^{c}\end{bmatrix}$$



The perspective camera model

- Note that $\mathbf{R}_{cw}$ and $\mathbf{t}_{cw}^{c}$ are the orientation and position of the world relative to the camera

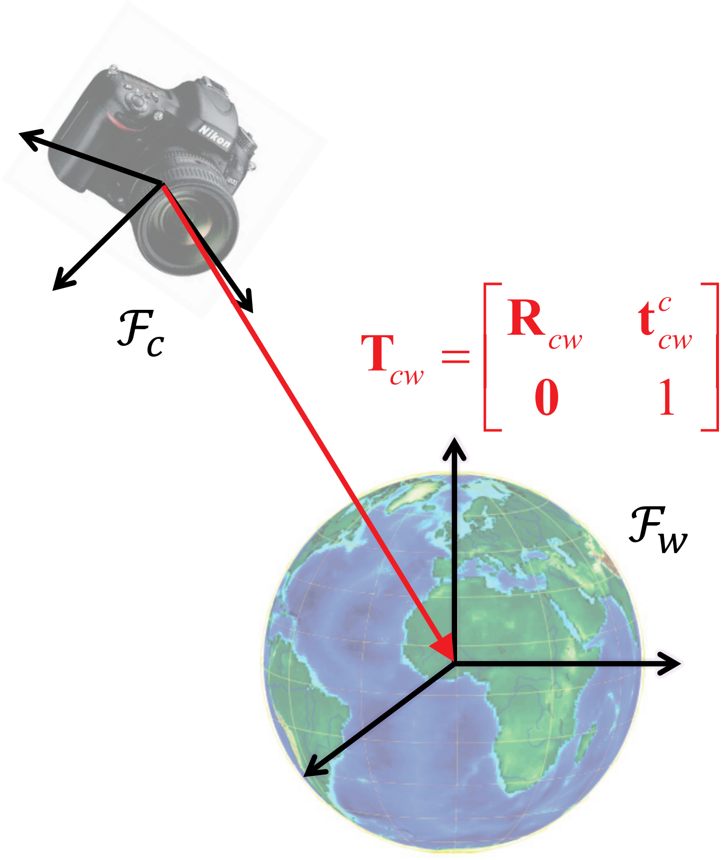
- It is more likely that we know the opposite

- Alternative formulation

$$\tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R}_{wc}^{T} & -\mathbf{R}_{wc}^{T}\mathbf{t}_{wc}^{w}\end{bmatrix}\tilde{\mathbf{x}}^{w}$$

where we have used that

$$\mathbf{T}_{cw} = \mathbf{T}_{wc}^{-1} = \begin{bmatrix}\mathbf{R}_{wc}^{T} & -\mathbf{R}_{wc}^{T}\mathbf{t}_{wc}^{w} \\ \mathbf{0} & 1\end{bmatrix}$$



Example

- This image was captured from a platform with a good onboard navigation system
- So we know the camera's position and orientation when the image was taken
- Based on what we now know about the perspective camera model and pose, we can
  - Project points in the scene into the image

Example

- This image was captured from a platform with a good onboard navigation system
- So we know the camera’s position and orientation when the image was taken
- Based on what we now know about the perspective camera model and pose, we can
  - Project points in the scene into the image

A terrain model projected into the image

Example

- This image was captured from a platform with a good onboard navigation system
- So we know the camera's position and orientation when the image was taken
- Based on what we now know about the perspective camera model and pose, we can
  - Project points in the scene into the image

Roads, railroad and a stream from a vector map

Example

- This image was captured from a platform with a good onboard navigation system
- So we know the camera's position and orientation when the image was taken
- Based on what we now know about the perspective camera model and pose, we can
  - Project points in the scene into the image
  - Backproject the image to the scene

Distance from the camera to the terrain model

Example

- This image was captured from a platform with a good onboard navigation system
- So we know the camera's position and orientation when the image was taken
- Based on what we now know about the perspective camera model and pose, we can
  - Project points in the scene into the image
  - Backproject the image to the scene

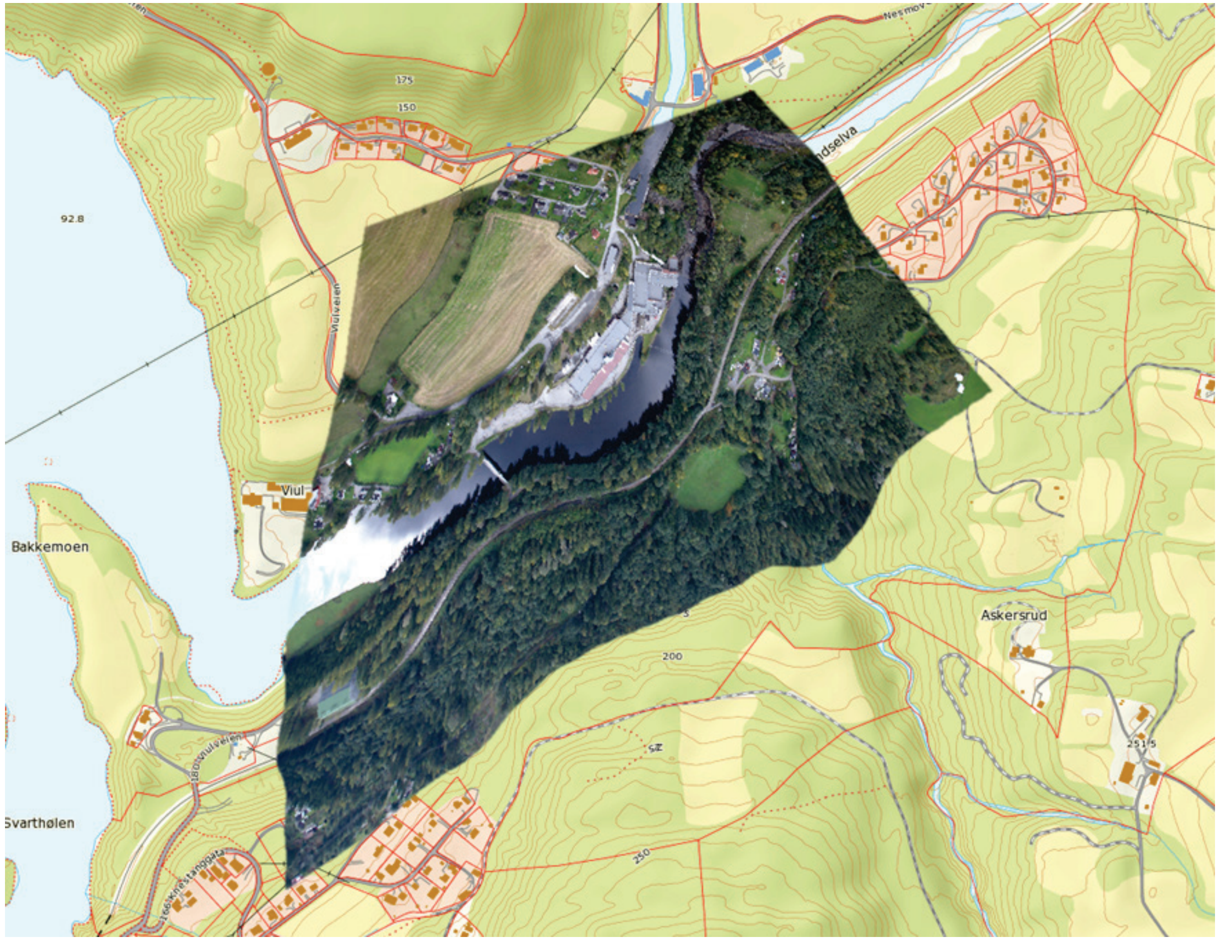
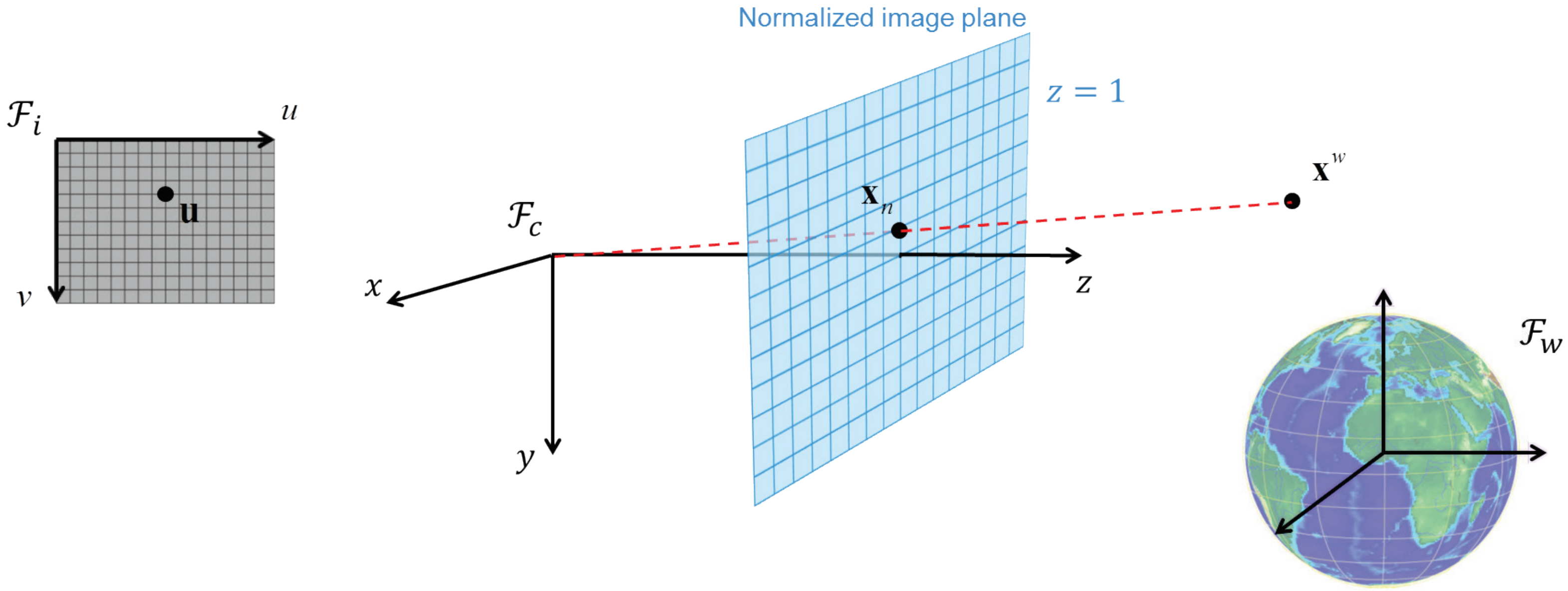


Image georeferenced by backprojection

Summary



- The perspective camera model when we consider 3D points in a frame $\mathcal{F}_w$ instead of the camera frame $\mathcal{F}_c$

$$\tilde{\mathbf{u}} = \mathbf{K}\begin{bmatrix}\mathbf{R}_{cw} & \mathbf{t}^c_{cw}\end{bmatrix}\tilde{\mathbf{x}}^w$$

Pose of $\mathcal{F}_w$ relative to $\mathcal{F}_c$!

$$\tilde{\mathbf{u}} = \underbrace{\begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{\Pi}_0} \underbrace{\begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}^c_{cw} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}_{cw}} \tilde{\mathbf{x}}^w$$

Further reading

- Do you want to know more?

- Online book by Richard Szeliski – Computer Vision: Algorithms and Applications
  http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf
  – Chapter 2 is about “image formation” and covers the perspective camera model in section 2.1.5

- Online book by Timothy D. Barfoot – State Estimation for Robotics
  http://asrl.utias.utoronto.ca/~tdb/bib/barfoot_ser17.pdf
  – Chapter 6.4 is about “sensor models” and covers the perspective camera model