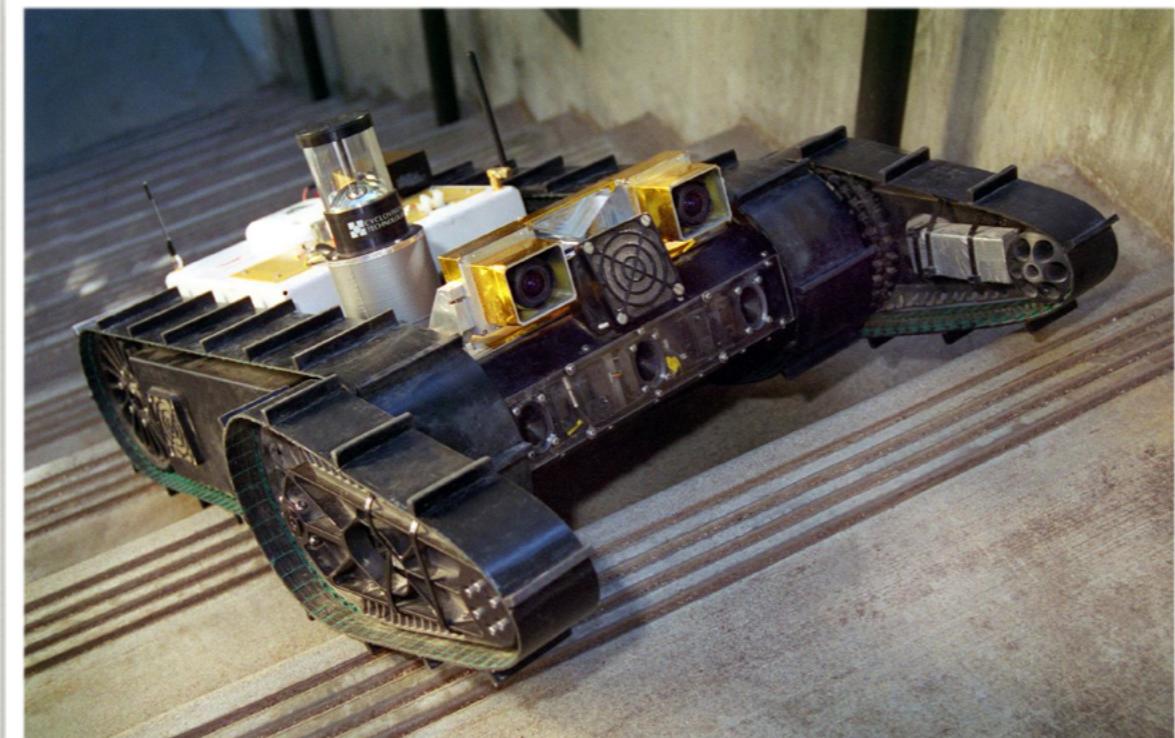
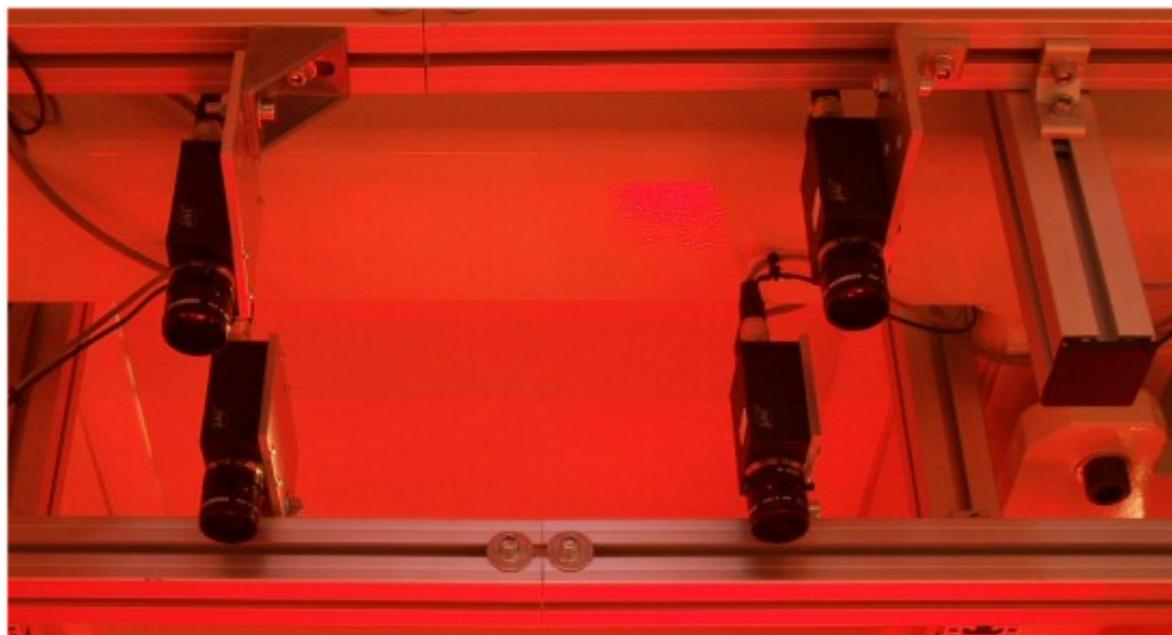


COMP3204/COMP6223: Computer Vision

Building machines that see

Jonathon Hare
jsh2@ecs.soton.ac.uk

Types of Computer Vision and their Environment



Scanning.

www.metmuseum.org/Cc

THE METROPOLITAN MUSEUM OF ART

Madame X (Madame Pierre Gautreau)
John Singer Sargent (American; Florence 1856–1925 London)

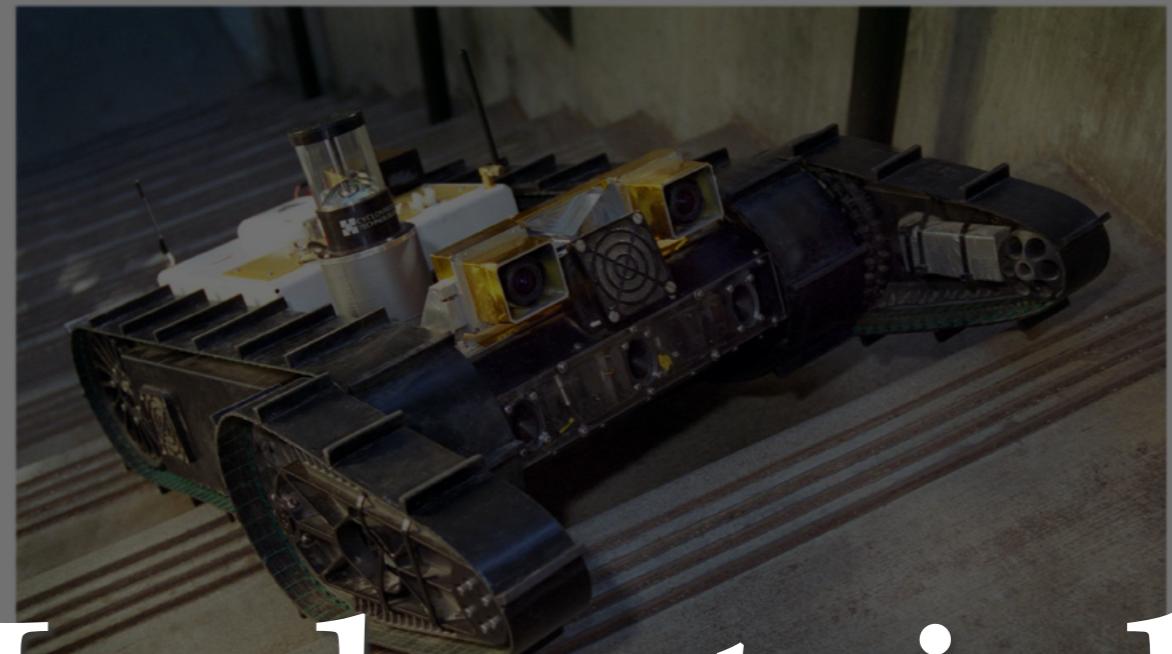
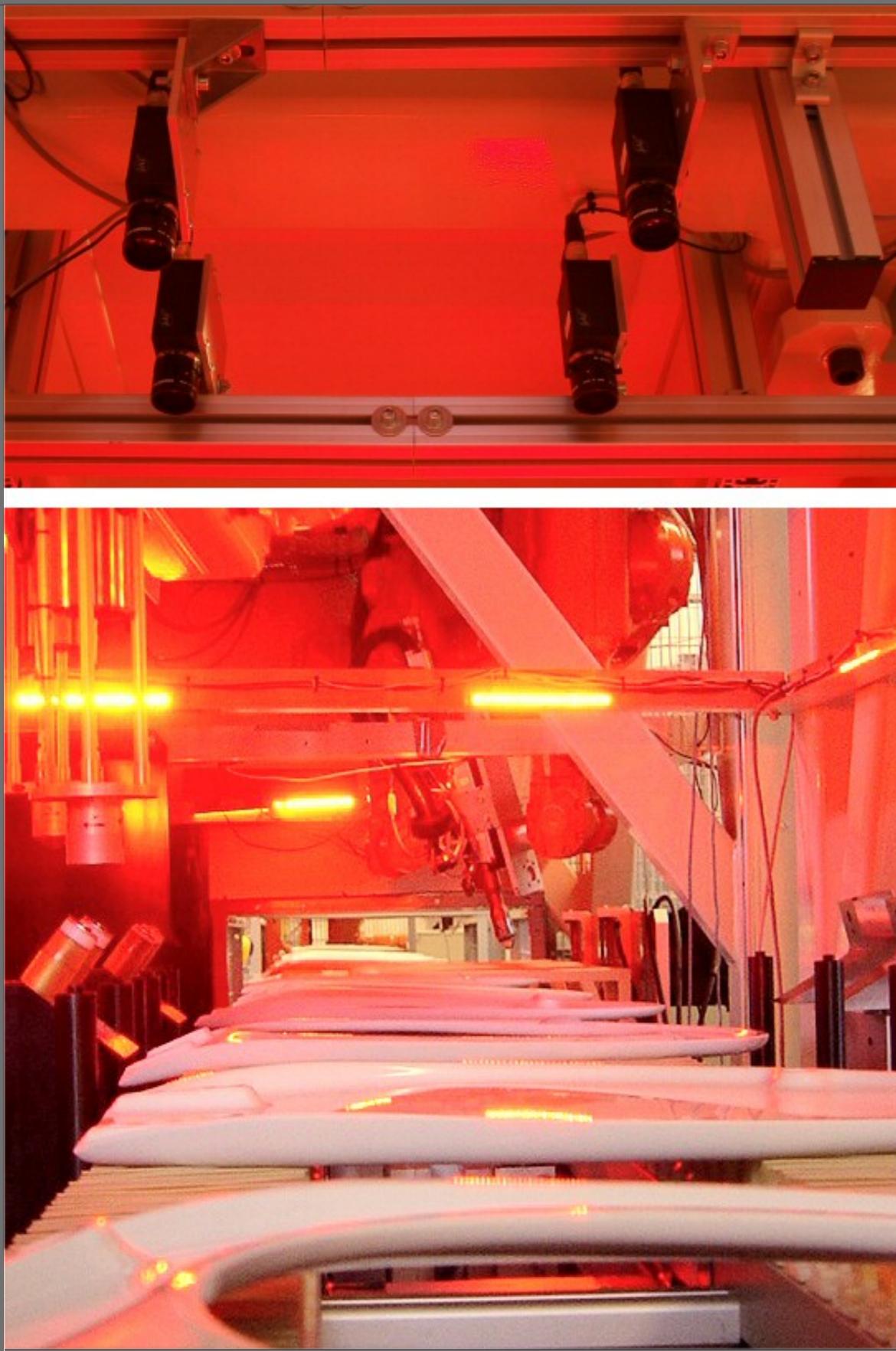
Enlarge Image

Date: 1883–84
Medium: Oil on canvas
Dimensions: 82 1/8 x 43 1/4 in. (208.6 x 109.9 cm)
Classification: Paintings
Credit Line: Arthur Hoppock Hearn Fund, 1916
Accession Number: 16.53
This artwork is not on display

+ Description

+ Signatures, Inscriptions, and

The screenshot shows a mobile browser displaying the Metropolitan Museum of Art's website. The main content is a thumbnail image of the painting "Madame X" by John Singer Sargent. A blue rectangular frame highlights a specific area of the painting. Below the image, the text "Scanning." is displayed. To the right, there is a detailed description of the artwork, including its title, artist, date, medium, dimensions, classification, credit line, accession number, and a note stating it is not on display. At the bottom, there are links for "Description" and "Signatures, Inscriptions, and".



Industrial Vision

Scanning.

www.metmuseum.org/Cc

M THE METROPOLITAN MUSEUM OF ART

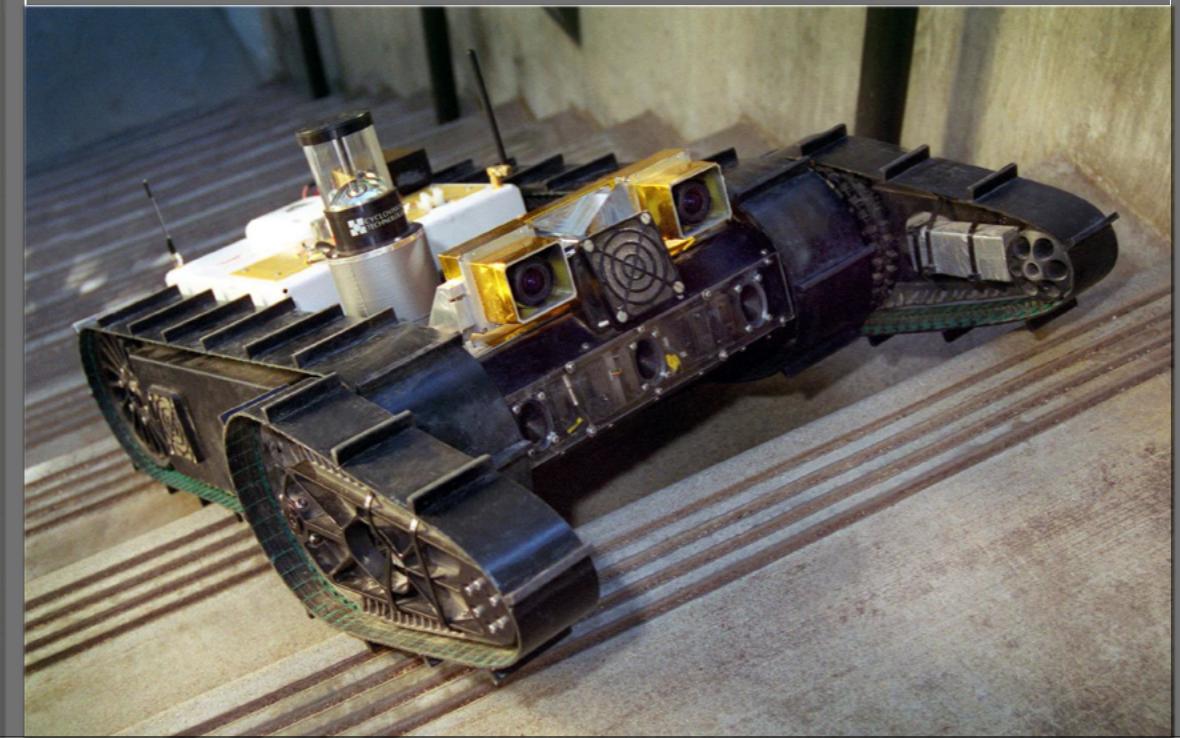
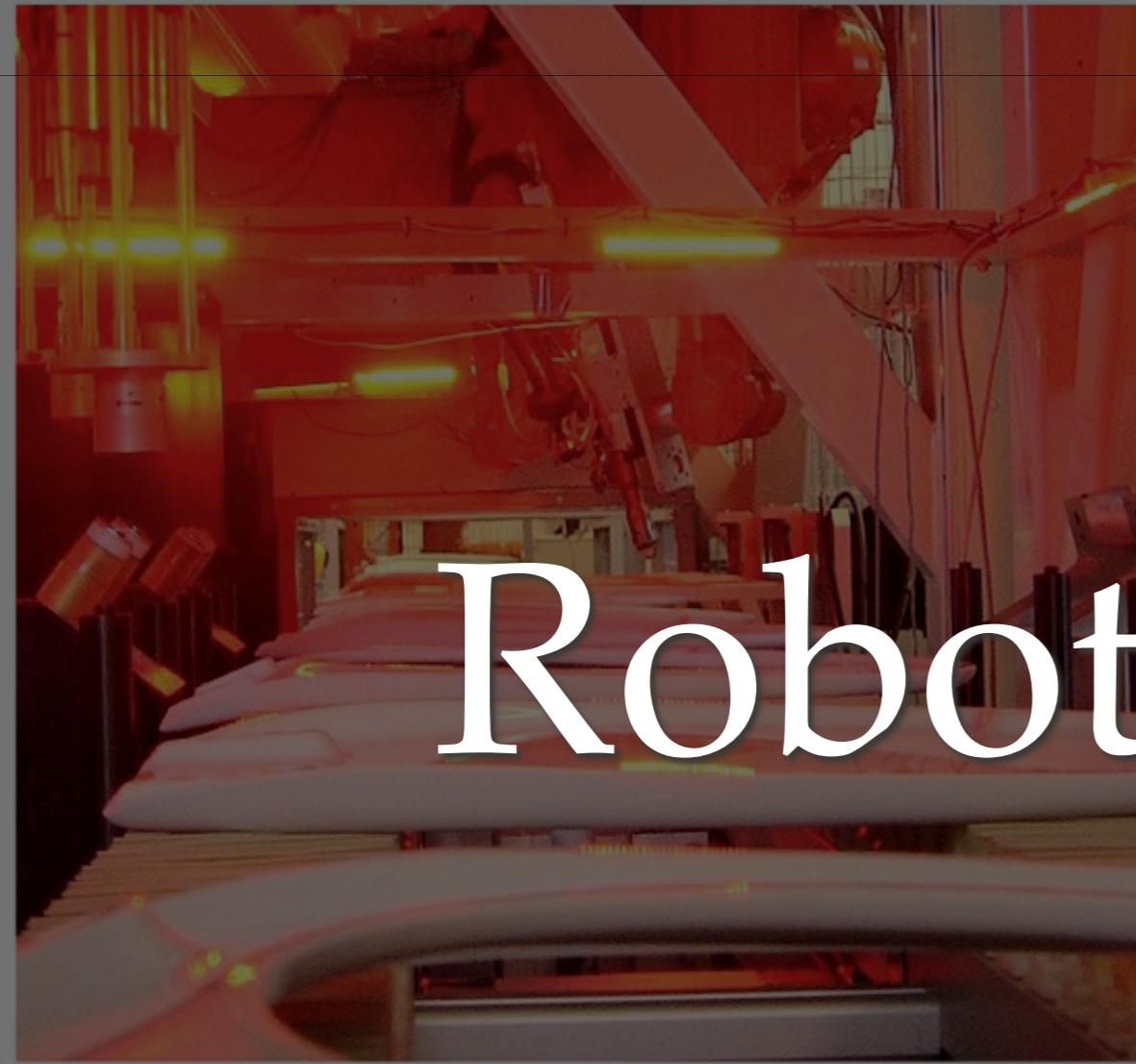
Madame X - Madame Pierre Gautreau

Date: 1883-84
Medium: Oil on canvas
Dimensions: 82 1/8 x 43 1/4 in. (208.6 x 109.9 cm)
Classification: Paintings
Credit Line: Arthur Hoppock Hearn Fund, 1916
Accession Number: 16.53
This artwork is not on display

+ Description

+ Signatures, Inscriptions, and

This image is a screenshot of a mobile device displaying a web page from the Metropolitan Museum of Art. The page shows the painting "Madame X" by John Singer Sargent. The title "Scanning." is visible at the top left. The main image of the painting is shown with a blue bounding box around the subject. Below the image, the painting's title and details are listed: Date: 1883-84, Medium: Oil on canvas, Dimensions: 82 1/8 x 43 1/4 in. (208.6 x 109.9 cm), Classification: Paintings, Credit Line: Arthur Hoppock Hearn Fund, 1916, Accession Number: 16.53. A note states "This artwork is not on display". At the bottom, there are links for "Description" and "Signatures, Inscriptions, and". The browser interface shows the URL "www.metmuseum.org/Cc" and the Met logo.



Robot Vision

Scanning.

www.metmuseum.org/Cc

THE METROPOLITAN MUSEUM OF ART

Madame X (Madame Pierre Gautreau)
John Singer Sargent (American, Florence 1856–1925 London)

Enlarge Image

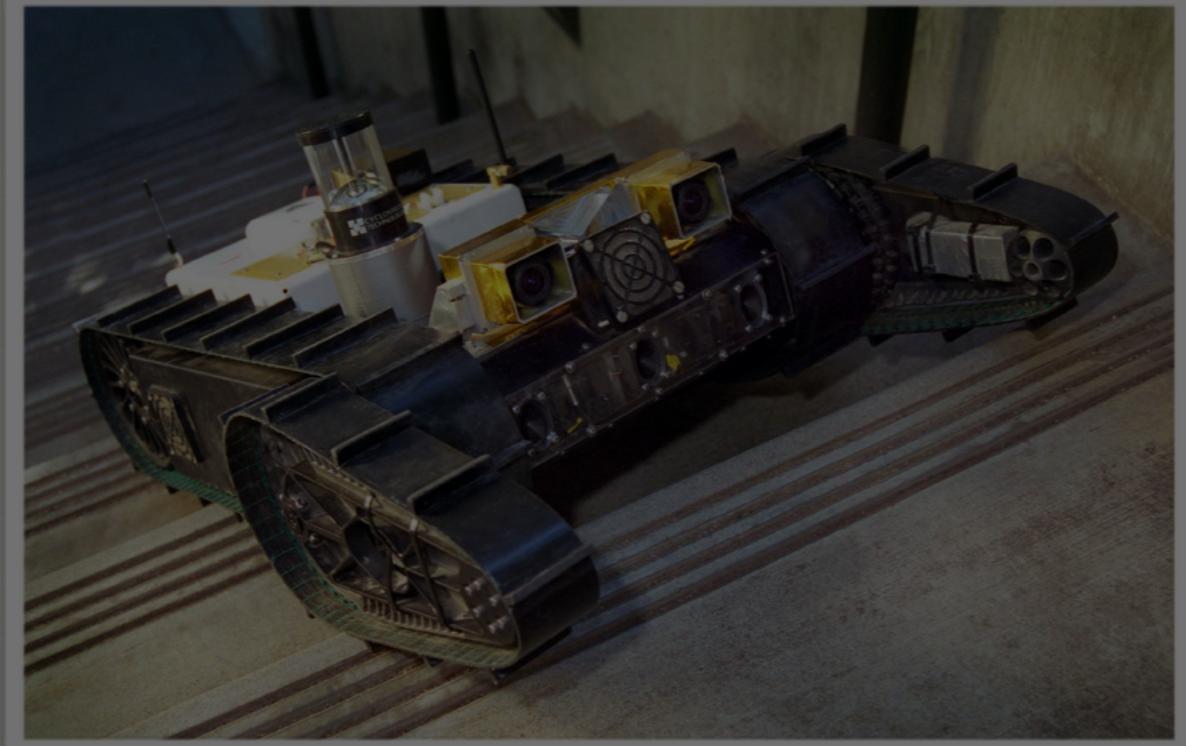
Date: 1884
Medium: Oil on canvas
Dimensions: 82 1/8 x 43 1/4 in. (208.6 x 109.9 cm)
Classification: Paintings
Credit Line: Arthur Hoppock Hearn Fund, 1916
Accession Number: 16.53
This artwork is not on display

+ Description

+ Signatures, Inscriptions, and

The image shows a screenshot of a mobile browser displaying the Metropolitan Museum of Art's website. The main content is a painting titled "Madame X" by John Singer Sargent, showing a woman in profile. The browser interface includes a URL bar with "www.metmuseum.org/Cc", a title bar with "THE METROPOLITAN MUSEUM OF ART", and a sidebar with details about the painting such as its date, medium, dimensions, and classification. Buttons for "Enlarge Image", "Description", and "Signatures, Inscriptions, and" are visible at the bottom.

Vision in the wild



Scanning.

www.metmuseum.org/Cc

THE METROPOLITAN MUSEUM OF ART

Madame X (Madame Pierre Gautreau)
John Singer Sargent (American; Florence 1856–1925 London)

Enlarge Image

Date: 1883–84
Medium: Oil on canvas
Dimensions: 82 1/8 x 43 1/4 in. (208.6 x 109.9 cm)
Classification: Paintings
Credit Line: Arthur Hoppock Hearn Fund, 1916
Accession Number: 16.53
This artwork is not on display

+ Description

+ Signatures, Inscriptions, and

A screenshot of a smartphone displaying a mobile version of the Metropolitan Museum of Art website. The page shows the painting "Madame X" by John Singer Sargent. The phone's status bar indicates the time is 1:41. The browser address bar shows the URL www.metmuseum.org/Cc. The main content area displays the painting and its details, including the date (1883–84), medium (Oil on canvas), dimensions (82 1/8 x 43 1/4 in. / 208.6 x 109.9 cm), classification (Paintings), credit line (Arthur Hoppock Hearn Fund, 1916), and accession number (16.53). A note states that the artwork is not on display. Below the main content, there are links for "Description" and "Signatures, Inscriptions, and".

What do all these systems have in
common?

Computer Vision Software

The screenshot shows the Eclipse IDE interface with two main windows. On the left is the Java editor window titled "Java - image-feature-extraction/src/main/java/org/openimaj/image/model/EigenImages.java - Eclipse SDK - /Users/jsh2/Documents/LMLK-Workspace". The code implements a Principal Component Analysis (PCA) for images, utilizing the OpenIMAJ library. It includes methods for extracting features from images, training a model on a list of images, reconstructing images from weight vectors, and visualizing individual principal components as images. On the right is a graphical user interface window titled "EigenImages" which displays 10 sliders labeled PC 0 through PC 9, each set to 0.00. A "Reset" button is located at the bottom of this window.

```
protected EigenImages() { }

/**
 * Construct with the given number of principal components.
 *
 * @param numComponents
 *          the number of PCs
 */
public EigenImages(int numComponents) {
    this.numComponents = numComponents;
    pca = new FeatureVectorPCA(new ThinSvdPrincipalComponentAnalysis(numComponents));
}

@Override
public DoubleFV extractFeature(FImage img) {
    final DoubleFV feature = FImage2DoubleFV.INSTANCE.extractFeature(img);
    return pca.project(feature);
}

@Override
public void train(List<? extends FImage> data) {
    final double[][] features = new double[data.size()][];
    width = data.get(0).width;
    height = data.get(0).height;

    for (int i = 0; i < features.length; i++)
        features[i] = FImage2DoubleFV.INSTANCE.extractFeature(data.get(i)).values;

    pca.learnBasis(features);
}

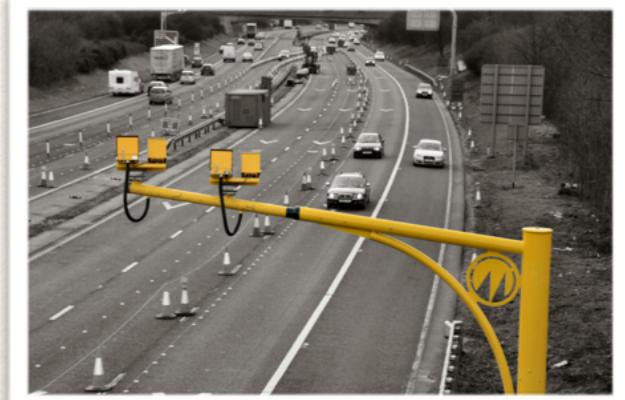
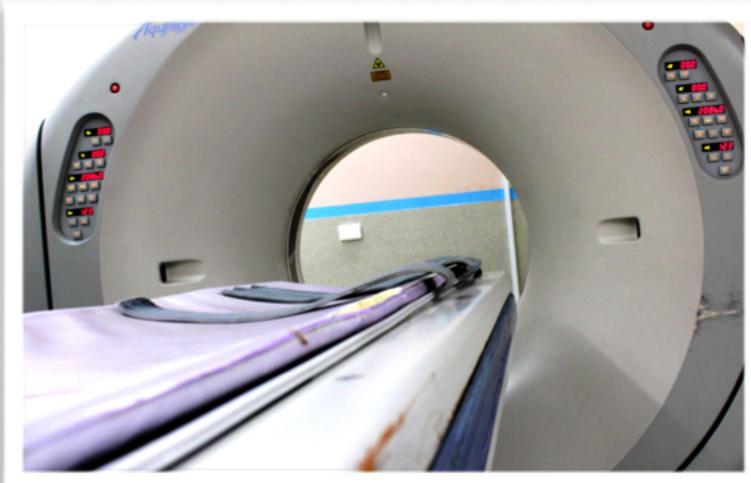
/**
 * Reconstruct an image from a weight vector
 *
 * @param weights
 *          the weight vector
 * @return the reconstructed image
 */
public FImage reconstruct(DoubleFV weights) {
    return DoubleFV2FImage.extractFeature(pca.generate(weights), width, height);
}

/**
 * Reconstruct an image from a weight vector
 *
 * @param weights
 *          the weight vector
 * @return the reconstructed image
 */
public FImage reconstruct(double[] weights) {
    return new FImage(ArrayUtils.reshapeFloat(pca.generate(weights)), width, height);
}

/**
 * Draw a principal component as an image. The image will be normalised so
 * it can be displayed correctly.
 *
 * @param pc
 *          the index of the PC to draw.
 * @return an image showing the PC.
 */
public FImage visualisePC(int pc) {
    return new FImage(ArrayUtils.reshapeFloat(pca.getPrincipalComponent(pc), width, height)).normalise();
}

@Override
```

Image Acquisition Hardware



but how do you go about designing
a computer vision system? and is
that all you need?

Key terms in designing CV systems

robust

invariant

repeatable

constraints



Key terms in designing CV systems

invariant

robust

repeatable

These are what you want

constraints



Key terms in designing CV systems

robust

invariant

repeatable

*This is what you design
your system to be*



Key terms in designing CV systems

robust

i *This is what you apply
to make it work* e

constraints



Robustness

- ❖ The vision system must be **robust** to changes in its environment
 - ❖ i.e. changes in lighting; angle or position of the camera; etc



Repeatability

- ❖ Repeatability is a *measure* of robustness
- ❖ Repeatability means that the system must work the same over and over, regardless of environmental changes



Invariance

- ❖ Invariance to environmental factors helps achieve robustness and repeatability
 - ❖ Hardware and software can be designed to be invariant to certain environmental changes
 - ❖ e.g. you could design an algorithm to be invariant to illumination changes...

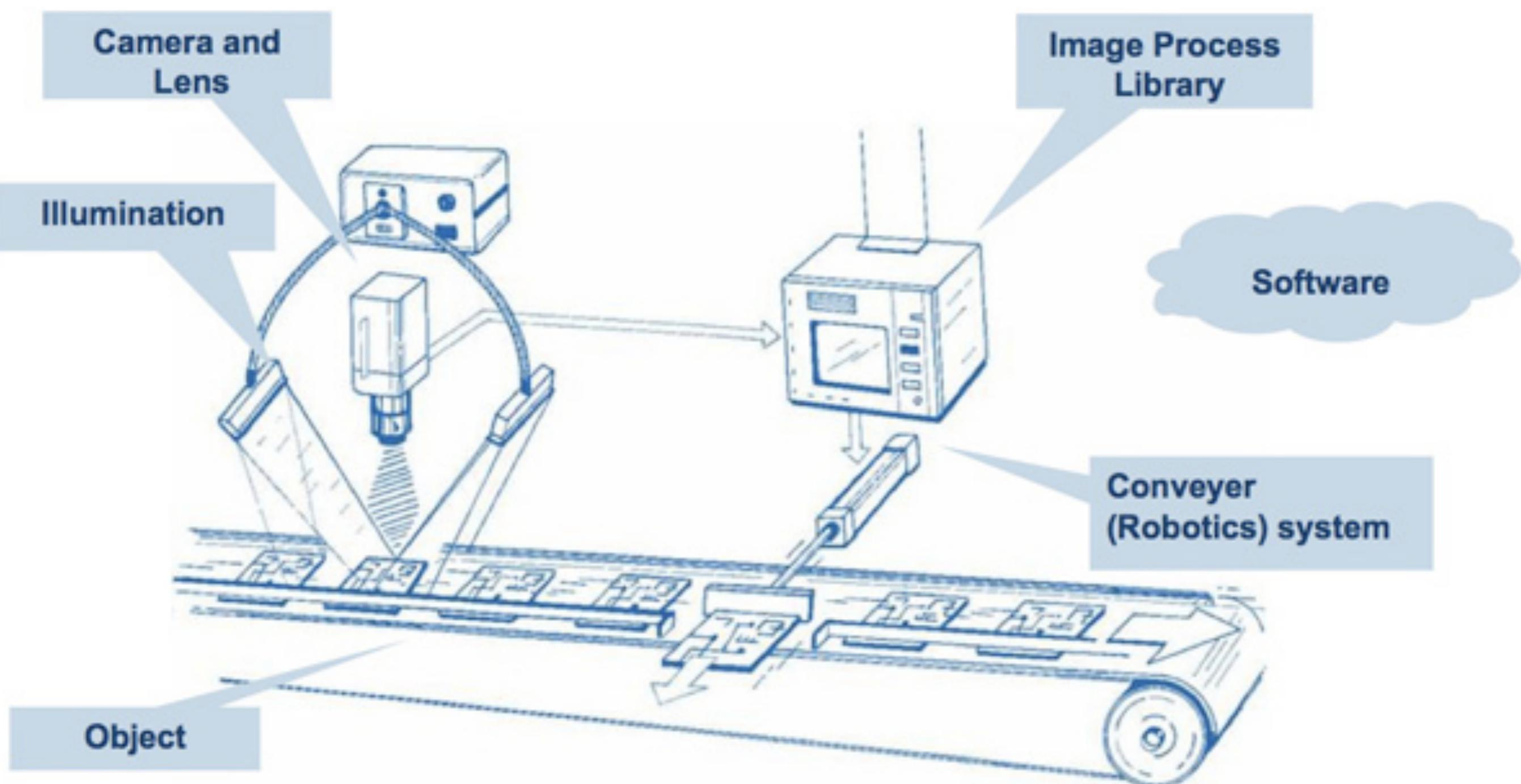


Constraints

- ❖ **Constraints** are what you apply to the hardware, software and wetware to make your computer vision system work in a repeatable, robust fashion.
- ❖ e.g. you constrain the system by putting it in a box so there can't be any illumination changes



Constraints in Industrial Vision



Software Constraints

- ❖ Really simple, but incredibly fast algorithms
 - ❖ Hough Transform is popular, but note that it isn't all that robust without physical constraints
 - ❖ Actually, same is true of most algorithms / techniques used in industrial vision
 - ❖ Intelligent use of colour...

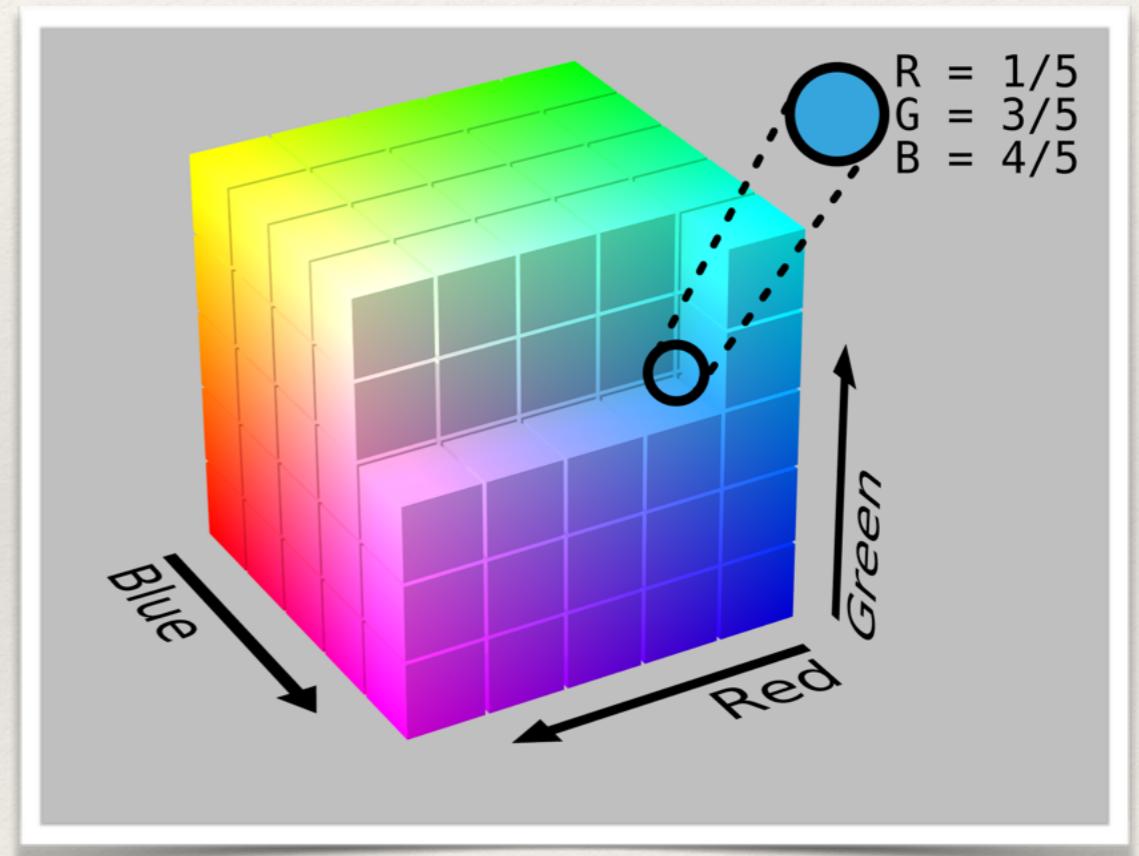


Important aside: Colour-spaces

- ❖ There are many different ways of *numerically* representing colour
 - ❖ A single representation of all possible colours is called a colour-space
 - ❖ It's *generally* possible to convert to one colour-space to another by applying a mapping (in the form of a set of equations or an algorithm)

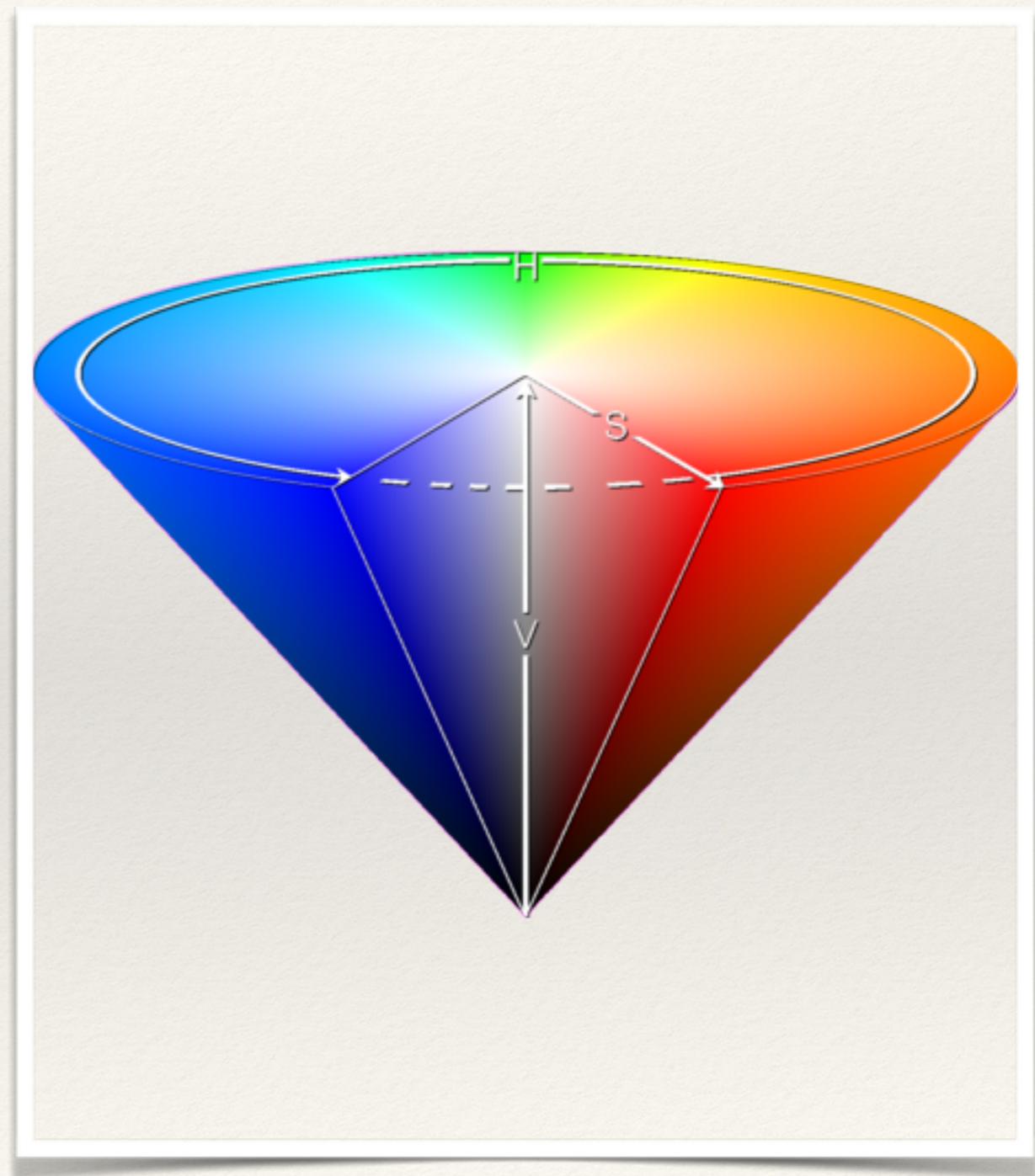
RGB Colour-space

- ❖ Most physical image sensors capture RGB
 - ❖ By far the most widely known space
 - ❖ RGB “couples” brightness (luminance) with each channel, meaning that illumination invariance is difficult.



HSV Colour-space

- ❖ Hue, Saturation, Value is another colour-space
 - ❖ Hue encodes the pure colour as an angle
 - ❖ **red == $0^\circ == 360^\circ$!!**
 - ❖ Saturation is how vibrant the colour is
 - ❖ And the Value encodes brightness
 - ❖ A simple way of achieving invariance to lighting is to use just the H or H & S components



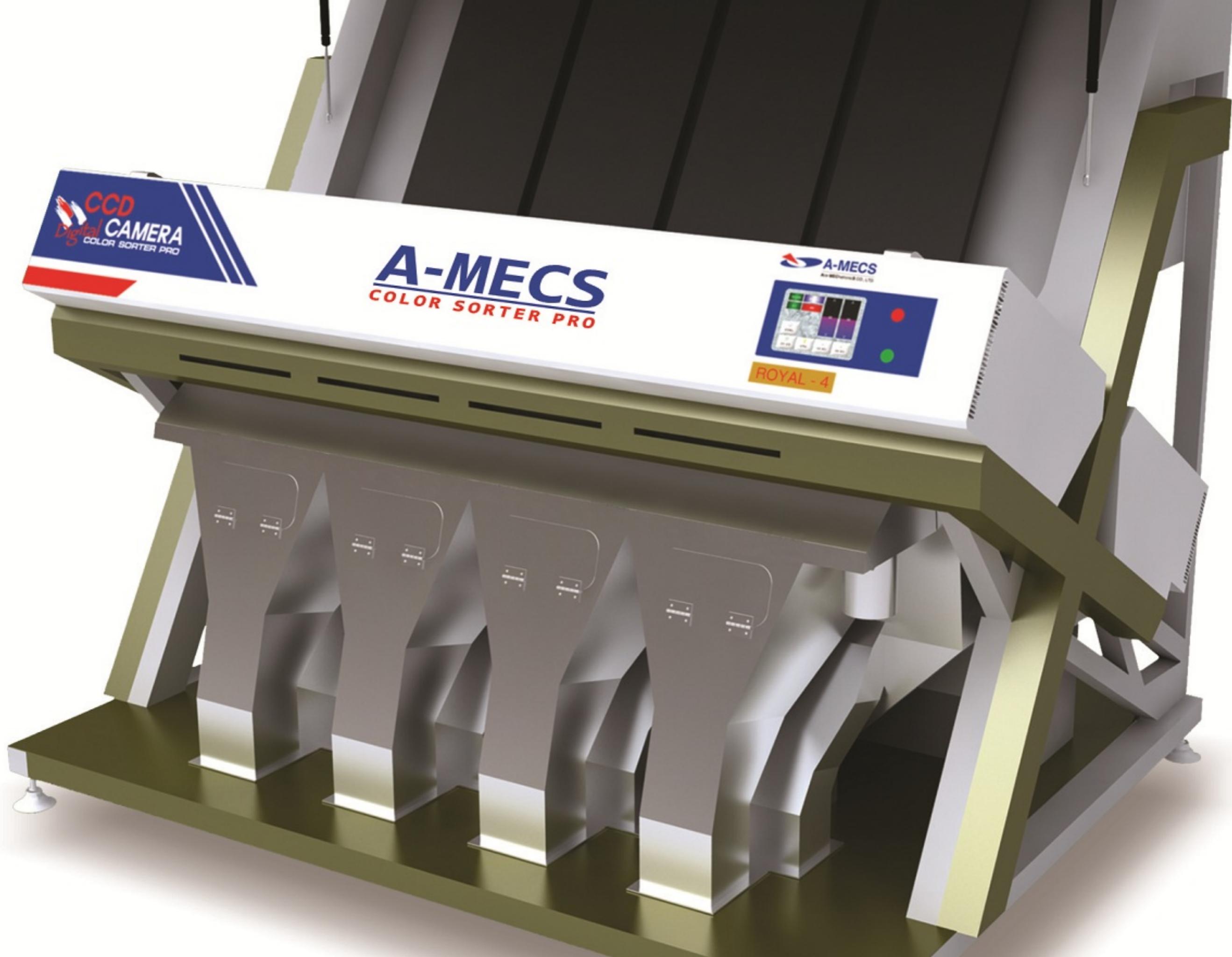
Demo: colour-spaces

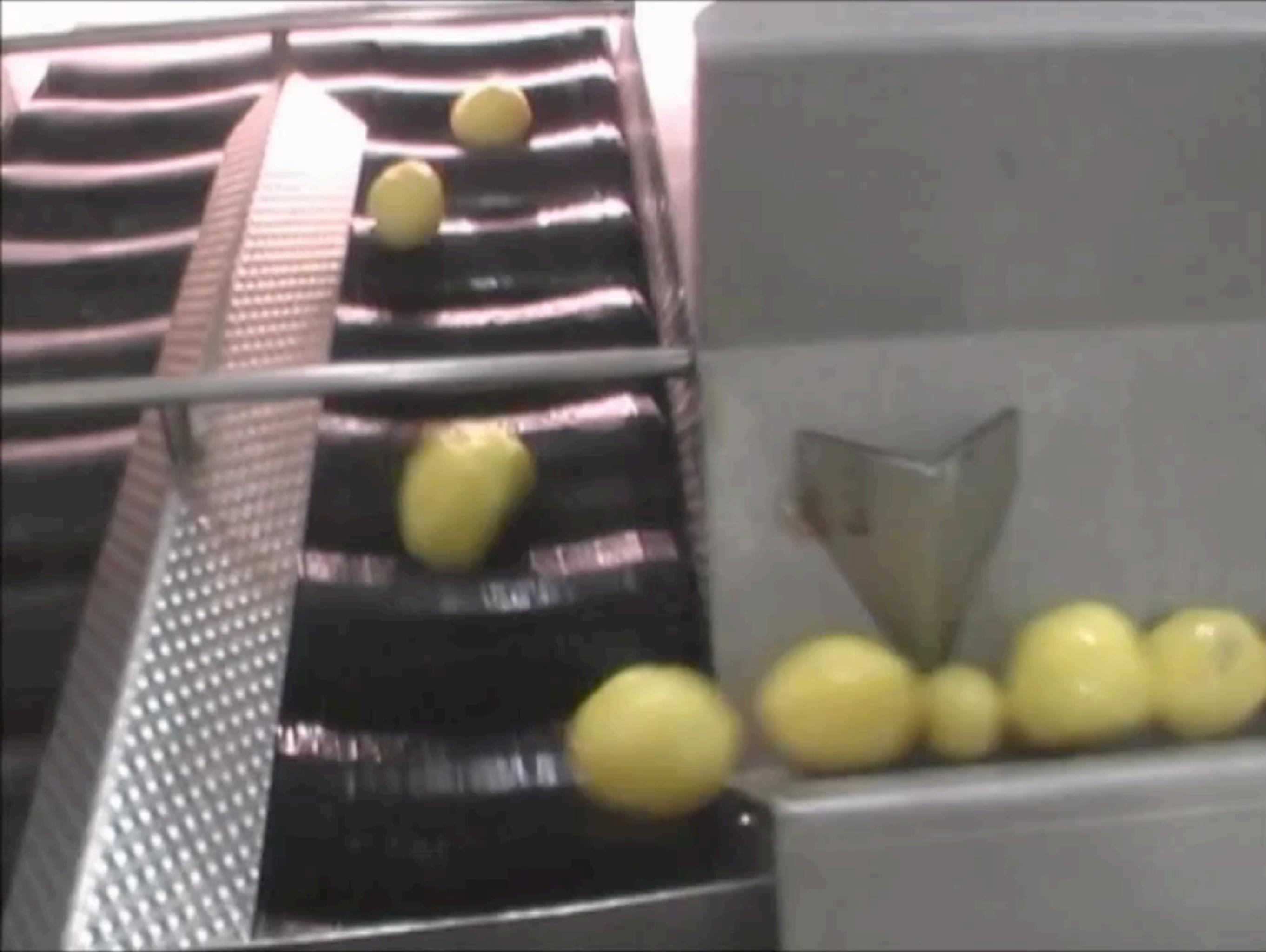
Physical Constraints

- ❖ Industrial vision is usually solved by applying simple computer vision algorithms, and lots of physical constraints:
 - ❖ Environment: lighting, enclosure, mounting
 - ❖ Acquisition hardware: expensive camera, optics, filters



*Let's look at some types of physical
constraint*





Vision in the wild

- ❖ So, what about vision systems in the wild, like ANPR cameras, or recognition apps for mobile phones?
 - ❖ Apply as many hardware and wetware constraints as possible, and let the software take up the slack
 - ❖ Colour information often less important than luminance



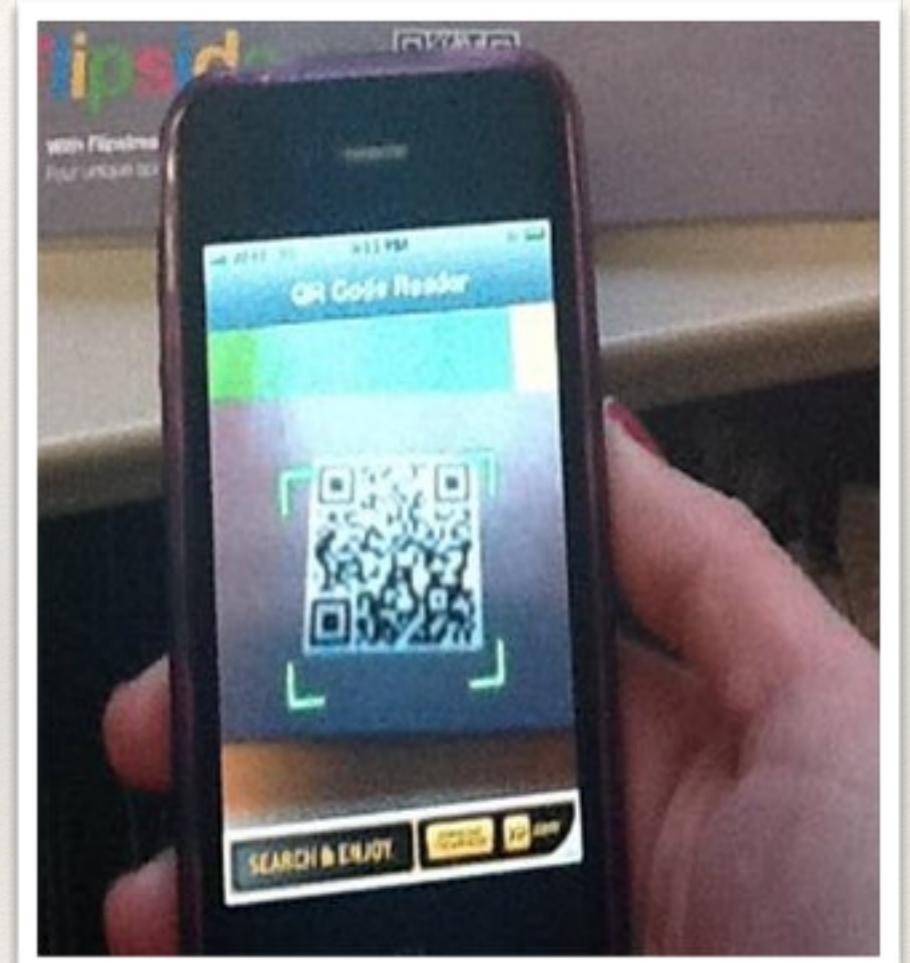
ANPR constraints

- ❖ License plate styles are different across the world, so most ANPR systems will only work with plates from a single country.
- ❖ License plates themselves are constrained in design:
 - ❖ Dimensions
 - ❖ Font
 - ❖ Material (IR reflectance!)



Mobile vision constraints

- ❖ QR-Codes are designed to be robust
- ❖ But most software requires (constrains) the user to operate in a certain way:
 - ❖ Orientation - approximately upright
 - ❖ Within a certain area
 - ❖ Approximately stationary



Almost unconstrained vision?

- ❖ As computers become more powerful, and new software techniques are developed to deal with invariance the need for constraints becomes less.
- ❖ ...but there is always going to be a problem of optimising the costs, and constraints can always help reduce costs

Summary

- ❖ **Robust** and **repeatable** computer vision is achieved through engineered **invariance** and applied **constraints**.