Lab 2
Energy efficient image processing

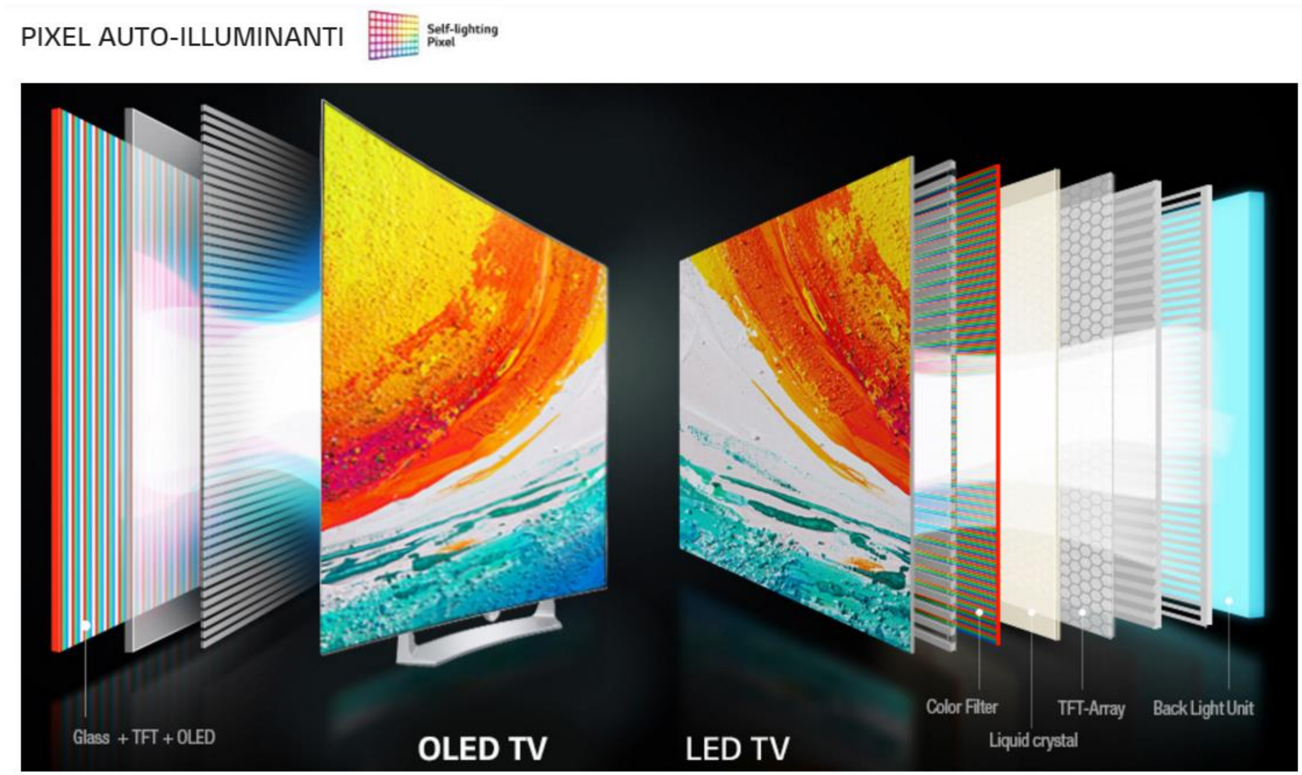
Objective and organization

- Demonstrates how manipulation of an image can be used to tradeoff image quality to save power in emissive displays
  - 1 report – 2 days
  - Matlab

- Organize all implemented methods in functions and scripts to **automatically** test and evaluate all images and all techniques

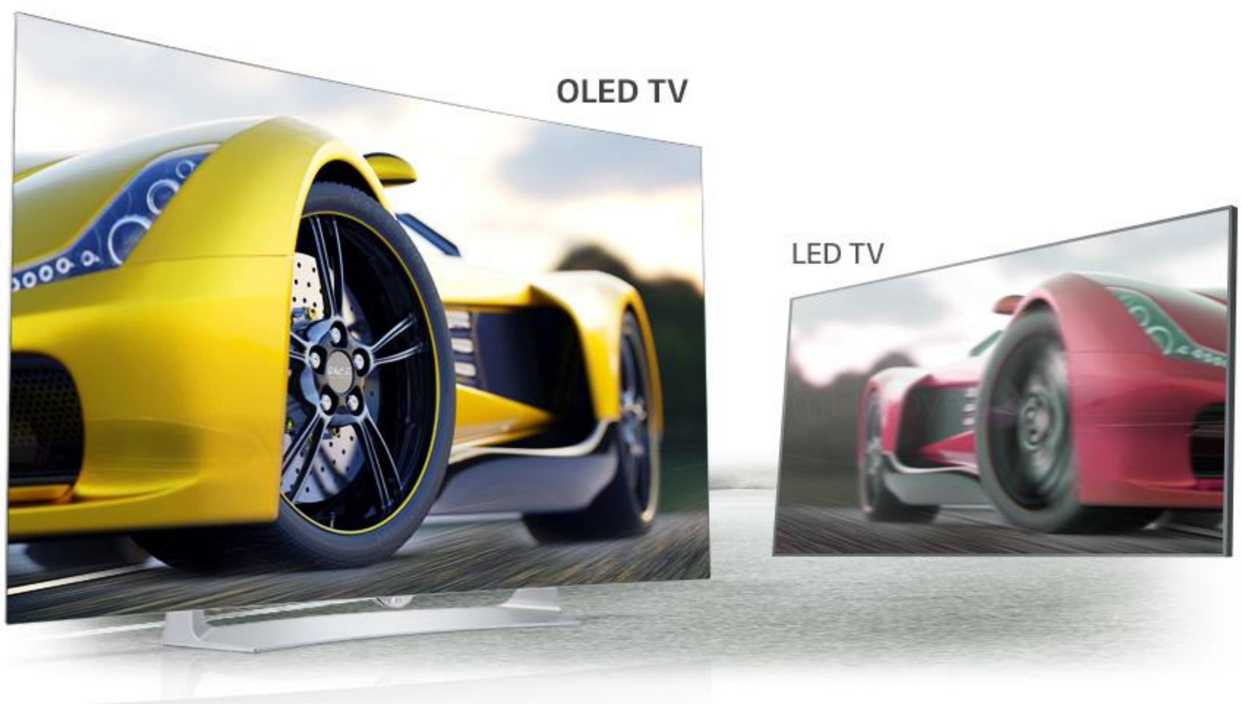
OLED vs LED

- OLED TVs
  - Do not require external lighting
    - Better black levels



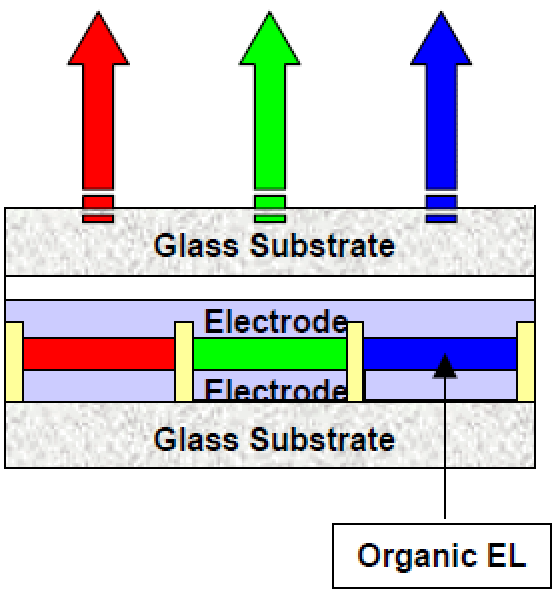
OLED vs LED

- OLED TVs
  - Pixels are independent from each other
    - More sense of depth
    - Higher contrast makes images more realistic



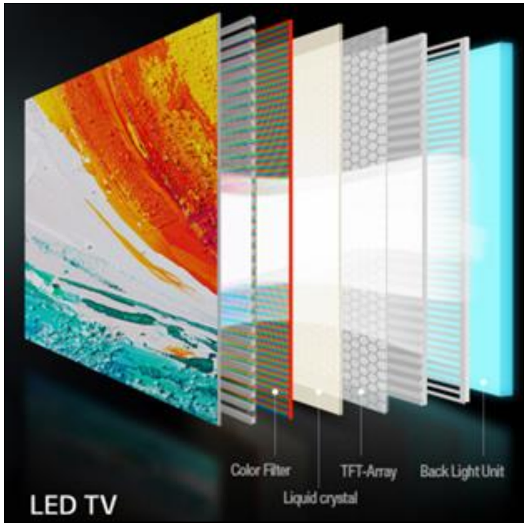
OLED

- Interesting case study from our perspective…
- Organic light-emitting diode (OLED)
  - Do not require external lighting
  - Pixels are emissive
    - Emissive layer is a film of organic compound which *emits light in response to an electric current*
  - Each pixel is made of three devices corresponding to red, green and blue components



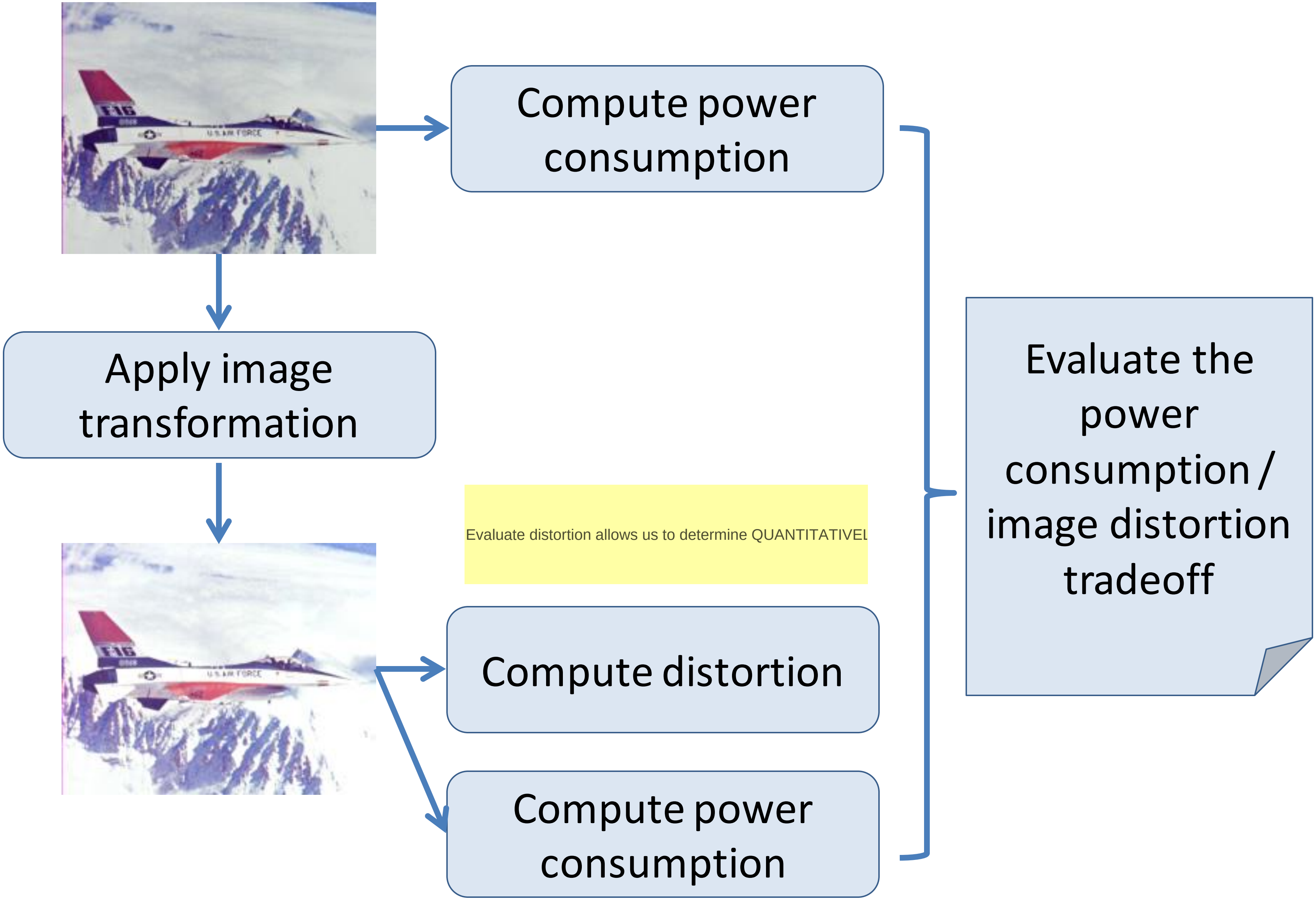
OLED

- In LCDs, backlight dominates power consumption and color has only negligible power impact
- With OLED displays, the color of a pixel impacts on power consumption
  - E.g., hungry blue
  - Different luminance efficacies
  - Different images imply different power consumption

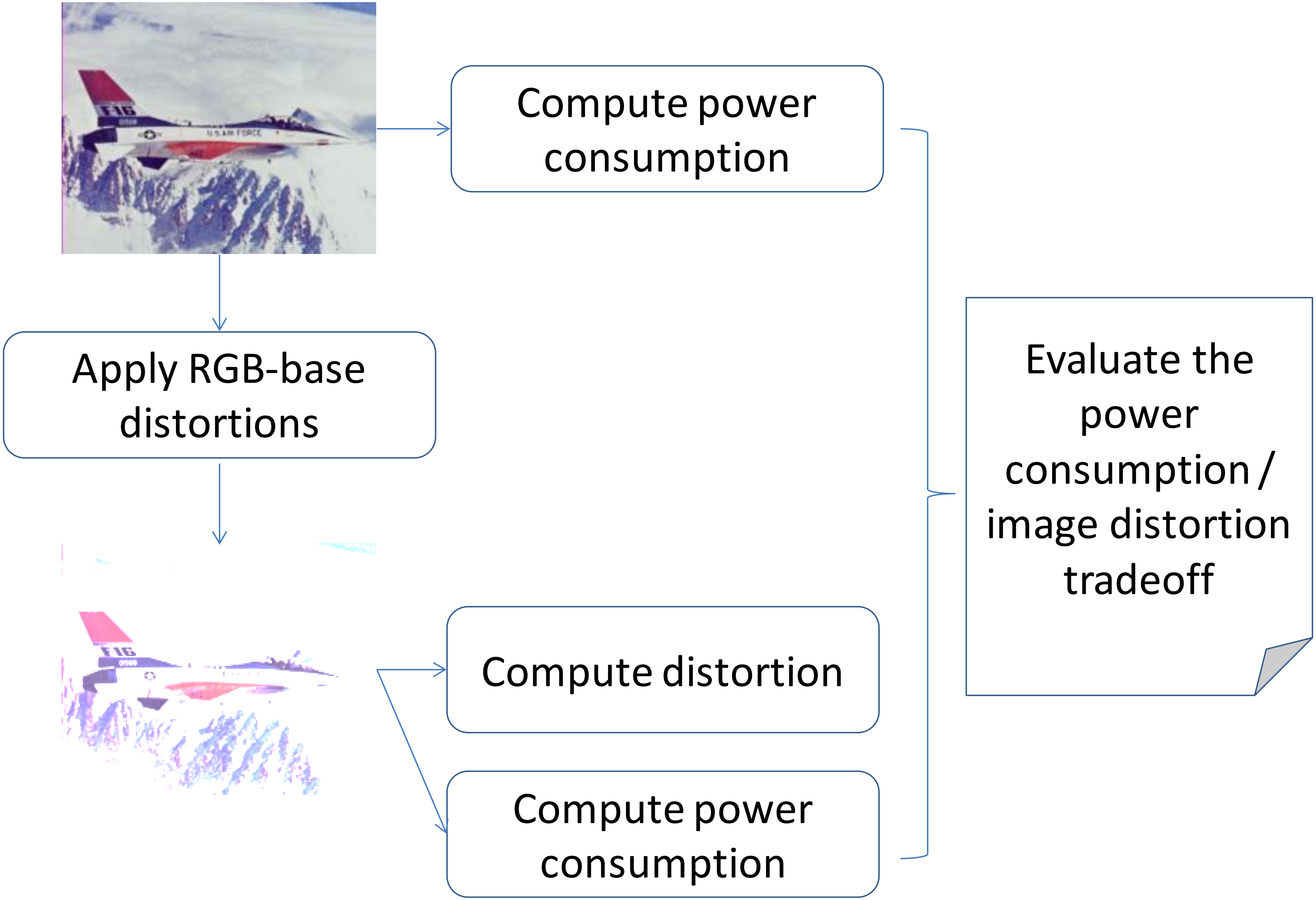


OLED

- Power consumption depends on color components of a pixel…
  - So we can save power by changing the spectrum of the image!
  - First class of power saving methods:
    - Change pixel color
    - Given a certain tolerance level on color distorsion

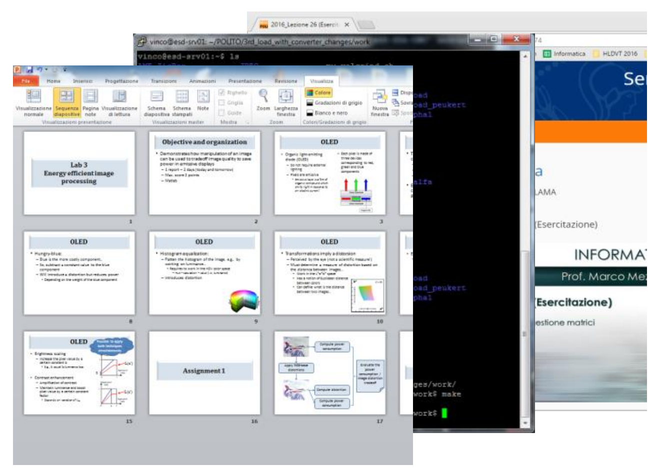

Compute power consumption
Apply image transformation
Evaluate distortion allows us to determine QUANTITATIVEL
Compute distortion
Compute power consumption
Evaluate the power consumption / image distortion tradeoff

Assignment 1


Compute power consumption
Apply RGB-base distortions
Compute distortion
Compute power consumption
Evaluate the power consumption / image distortion tradeoff

1. Identification of images

- Test images will be:
  - 10 color images from the USC SIPI database
    - http://sipi.usc.edu/database/database.php?volume=misc
  - 5 images representing different screenshots of your computer
- Different colors and characteristics…



2. Manipulation of images

- Experiments require to adopt different color spaces…
- **TASK:** Learn how to:
  - Import the image
    - imread() function
  - Extract the R, G, B channels
  - Convert between different color spaces
- Refer to:
  - http://it.mathworks.com/help/images/index.html

Documentation

Contents

- Documentation
  - Image Processing Toolbox
    - Getting Started with Image Processing Toolbox
    - Image Processing Toolbox Examples
    - Release Notes
    - Functions
    - Classes
    - Apps
    - Import, Export, and Conversion
    - Display and Exploration
    - Geometric Transformation, Spatial Referencing, and Image Registration
    - Image Enhancement
    - Image Analysis
    - Color
    - Code Generation
    - GPU Computing

Image Processing Toolbox

R2014b

Perform image processing, analysis, and algorithm development

Getting Started | Examples | Release Notes

Import, Export, and Conversion

Image data import and export, conversion of image types and classes

Display and Exploration

Interactive tools for image display and exploration

Geometric Transformation, Spatial Referencing, and Image Registration

Scale, rotate, perform other N-D transformations, provide spatial information, align images using automatic or control point registration

Image Enhancement

Contrast adjustment, morphological filtering, deblurring, and other image enhancement tools

Image Analysis

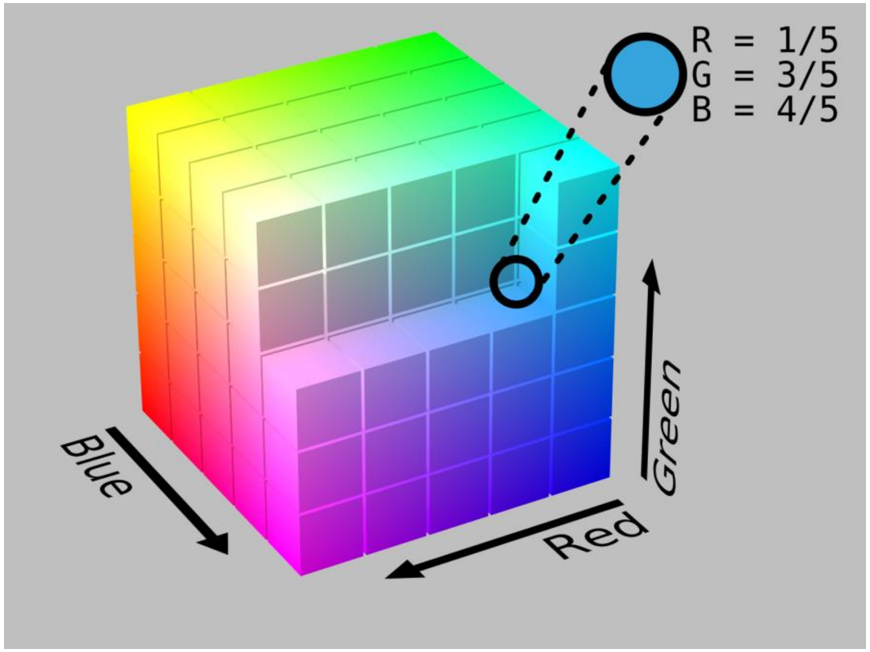
Region analysis, texture analysis, pixel and image statistics

Color

Color space conversions, support for International Color Consortium (ICC) profiles

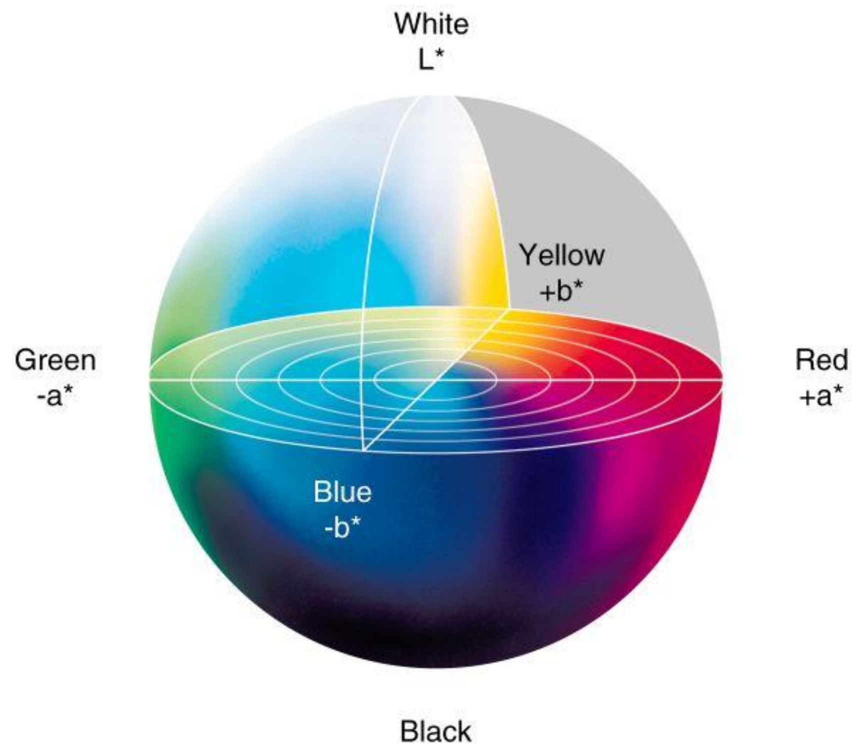
2. Manipulation of images

- RGB
  - Additive color space
    - All possible colors that can be made from three colorants for red, green and blue
  - Stores individual values for red, green and blue
  - Convenient color model for computer graphics as it is similar to the human visual system
    - Used in LCDs



2. Manipulation of images

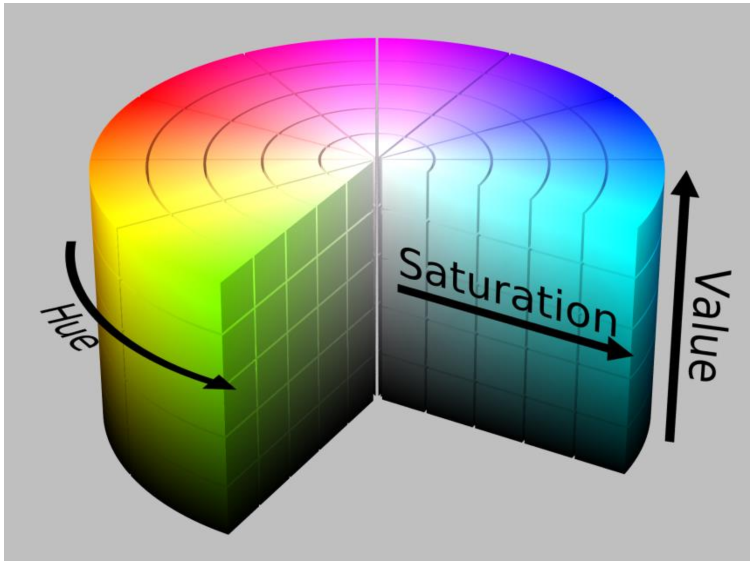
- Lab
  - One channel for luminance (L) and two color channels (a and b)
  - Includes all perceivable colors
    - Super-set of RGB
  - The space is a three-dimensional Real number space
    - Allows the definition of Euclidean distance

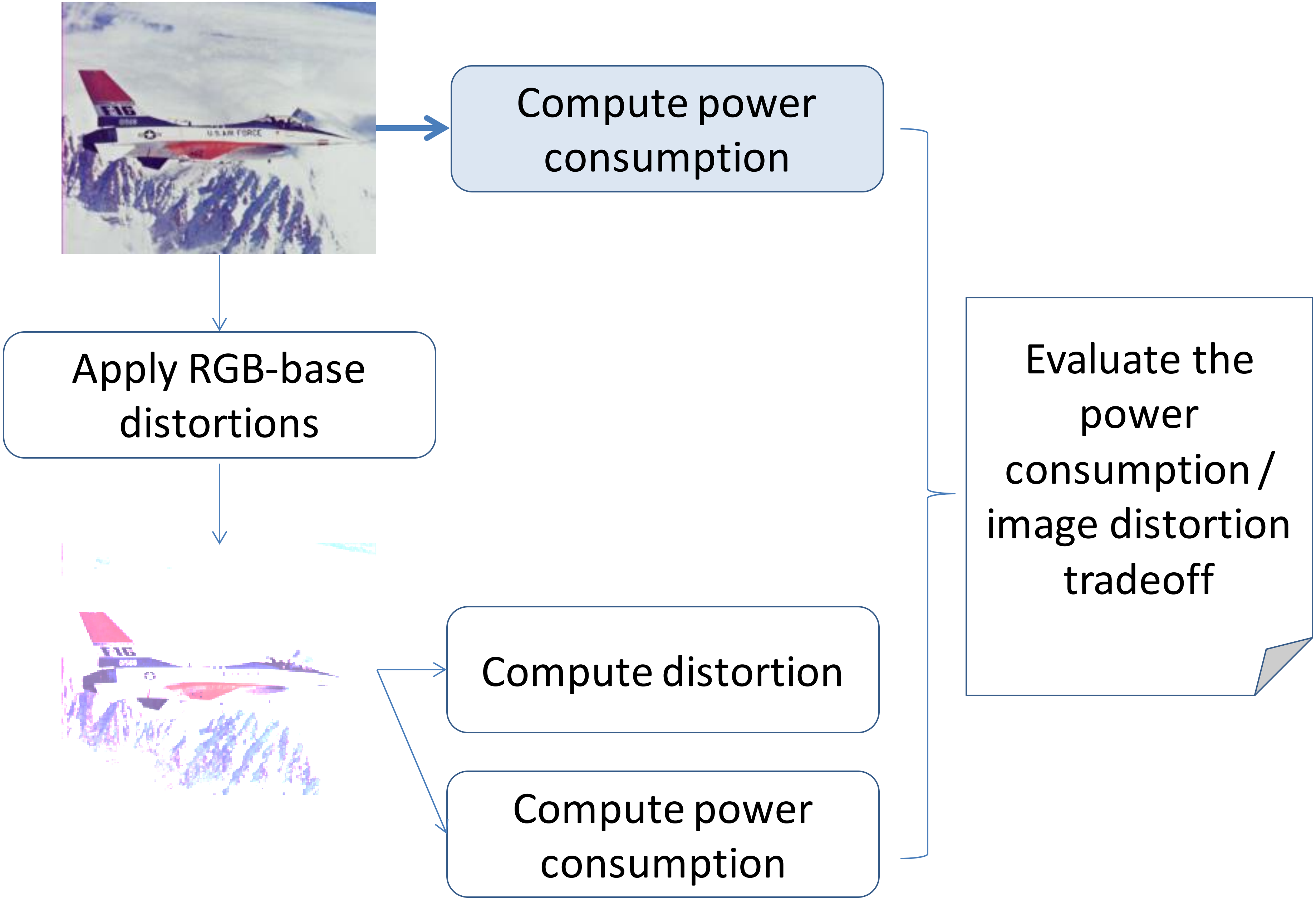


2. Manipulation of images

There are functions in matlab used to move from

- HSV
  - Hue
    - Perceived color
  - Saturation
    - Colorfulness, amount of white component
  - Value
    - Brightness
  - Cylindrical-coordinate representations of points in an RGB color model
  - Widely used in computer graphics




Compute power consumption
Apply RGB-base distortions
Compute distortion
Compute power consumption
Evaluate the power consumption / image distortion tradeoff

3. Evaluation of power consumption

Compute the power for a given image.

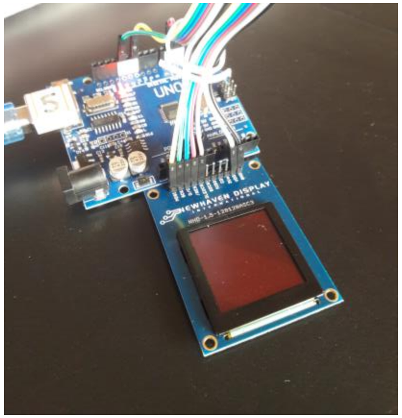
- Power model
  - $P_{pixel} = f(R) + h(G) + k(B)$
    - Depends on pixel color in terms of RGB components
    - f, h and k determined experimentally by:
      - Setting black screen to estimate C
      - For f, set G and B components to 0 and vary R component
      - Similar for h and k

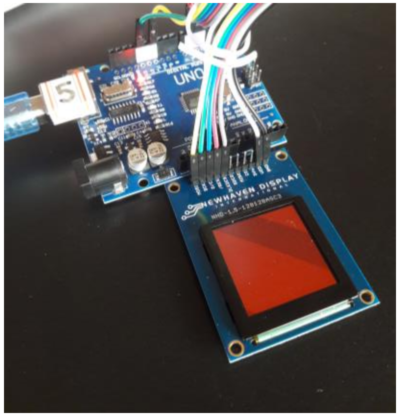
Total power = sum power fo all the pixels

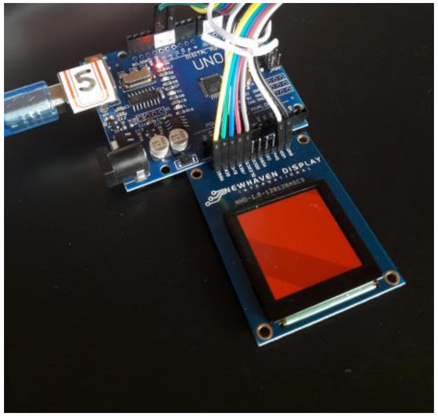
  - $P_{image} = C + \sum_{i=1}^{n}\{f(R_i) + h(G_i) + k(B_i)\}$
    - Sums up power contributions of single pixels
    - C static power independent of pixel values

1. Evaluation of power consumption

- Experimental Setup:
  - Show monochromatic images with different RGB values on the OLED, e.g.:





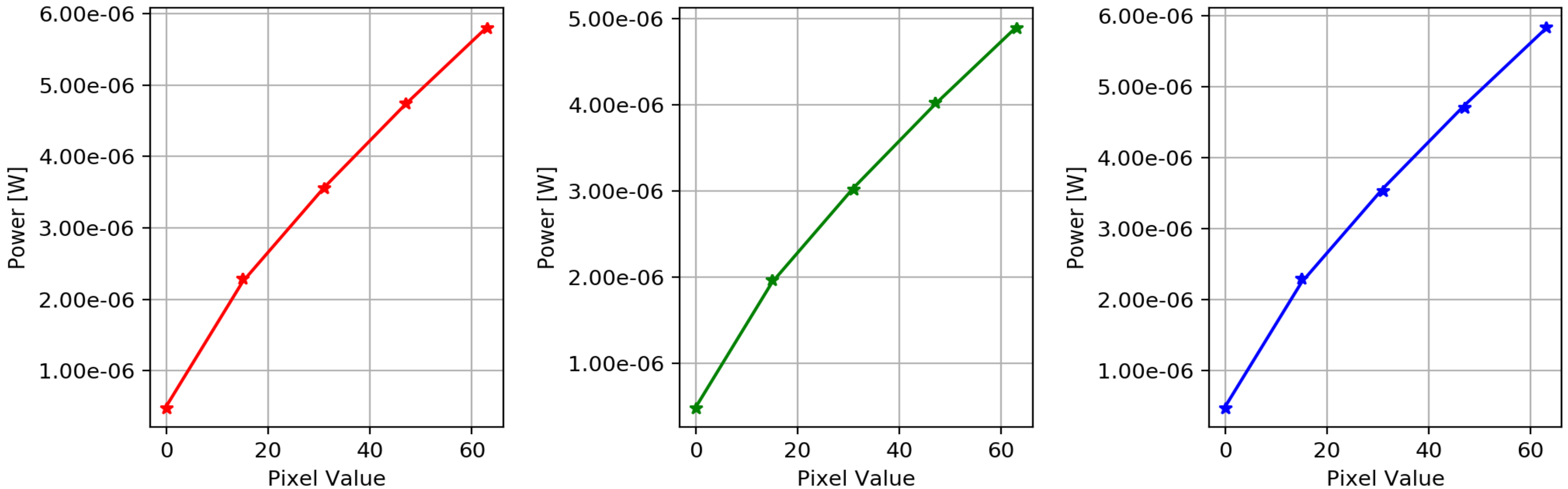


  - Measure power supply current (and convert to power)

1. Power model for the provided OLED (cont'd)

- Interpolation:
  - Find regression model type that fits best the data and determine the corresponding parameters

6 bits per channel (RGB565?), so that's wh ythe pixel va

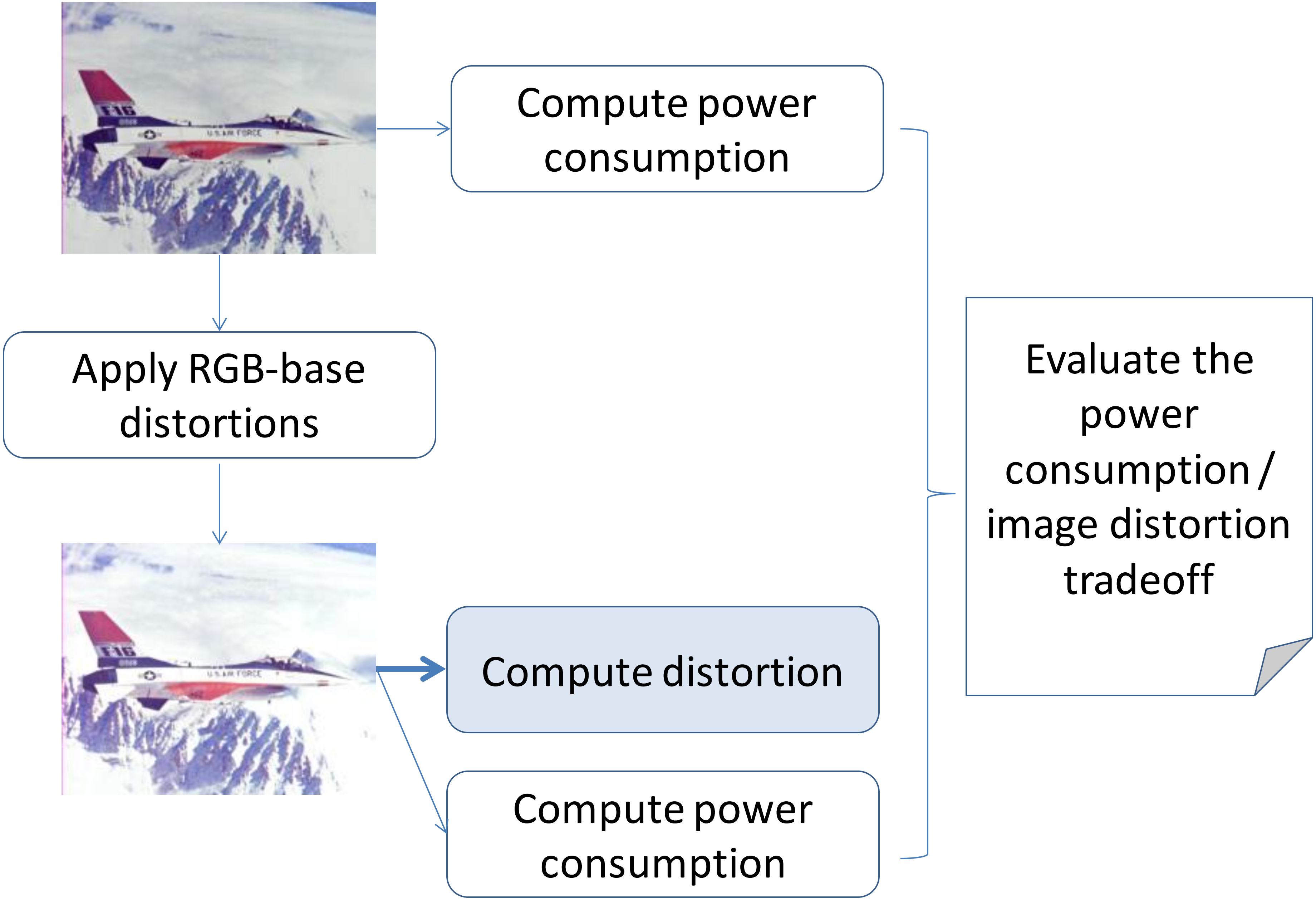


  - For this Lab, we used a model format from literature (see next slide)

3. Evaluation of power consumption

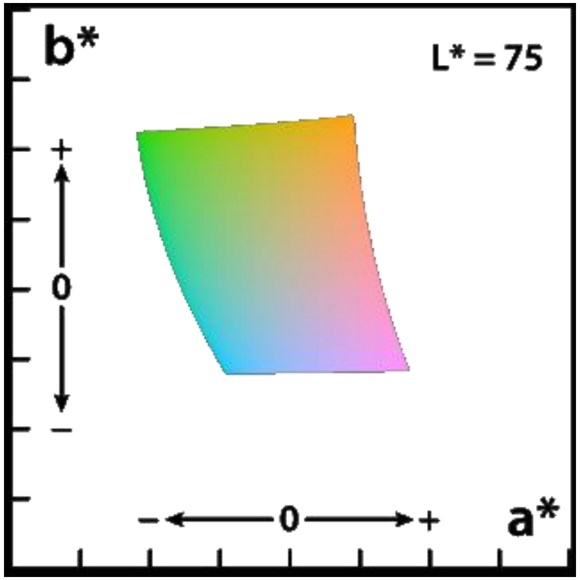
- **TASK:** Define a MATLAB function that estimates power consumed to display an image
  - $P_{pixel} = w_R * R^{\gamma} + w_G * G^{\gamma} + w_B * B^{\gamma}$
  - $P_{image} = w_0 + \sum_{i=1}^{n}\{P_i(R, G, B)\}$
  - R, G, B are pixel values between 0 and 255

γ	W_0	W_R	W_G	W_B
0.7755	$1.48169521*10^{-6}$	$2.13636845*10^{-7}$	$1.77746705*10^{-7}$	$2.14348309*10^{-7}$


Compute power consumption
Apply RGB-base distortions
Evaluate the power consumption / image distortion tradeoff
Compute distortion
Compute power consumption

4. Evaluation of image distortion

- Transformations imply a distorsion
  - Perceived by the eye (not a scientific measure!)
  - Must determine a measure of distortion based on the *distance* between images…
    - Work in the L*a*b* space
    - Has a notion of Euclidean distance between colors
    - Can define what is the distance between two images..



Everytime we transform an image we are applying a distortion: 3 measurements:* Euclidean dista

4. Evaluation of image distortion

- Evaluation of image distortion
  - Difference between two images
    - $\varepsilon\left(image_i, image_j\right) =$

Total distortion between t

$$\sum_{k=1}^{N}\left(\sqrt{\left(\left(L_{i,k} - L_{J,k}\right)^2\right) + ((a_{i,k} - a_{J,k})^2) + ((b_{i,k} - b_{J,k})^2)}\right)$$

    - N = number of pixels
    - k = kth bit
    - Bit per bit, compute the difference of L, a and b components between the two images

4. Evaluation of image distortion

- **TASK:** Define a MATLAB function that estimated the distortion w.r.t. the original image
  - $\varepsilon\left(image_i, image_j\right) =$

$$\sum_{k=1}^{N}\left(\sqrt{\left(\left(L_{i,k} - L_{J,k}\right)^2\right) + ((a_{i,k} - a_{J,k})^2) + ((b_{i,k} - b_{J,k})^2)}\right)$$

  - Work in the L*a*b* space and compute the Euclidian distance pixel per pixel
  - Convert by using MATLAB's r*gb2lab()* and *lab2rgb()* functions

4. Evaluation of image distortion

- Think also in terms of percentage distortion
  - E.g., distortion of new image w.r.t. distortion between a black and a white image
  - $dist = \frac{\varepsilon(image_{new}, image_{orig})}{\varepsilon(white\ image, black\ image)} \cdot 100 \qquad (\%)$

W * H * ($100^2 + 255^2 + 255^2$)

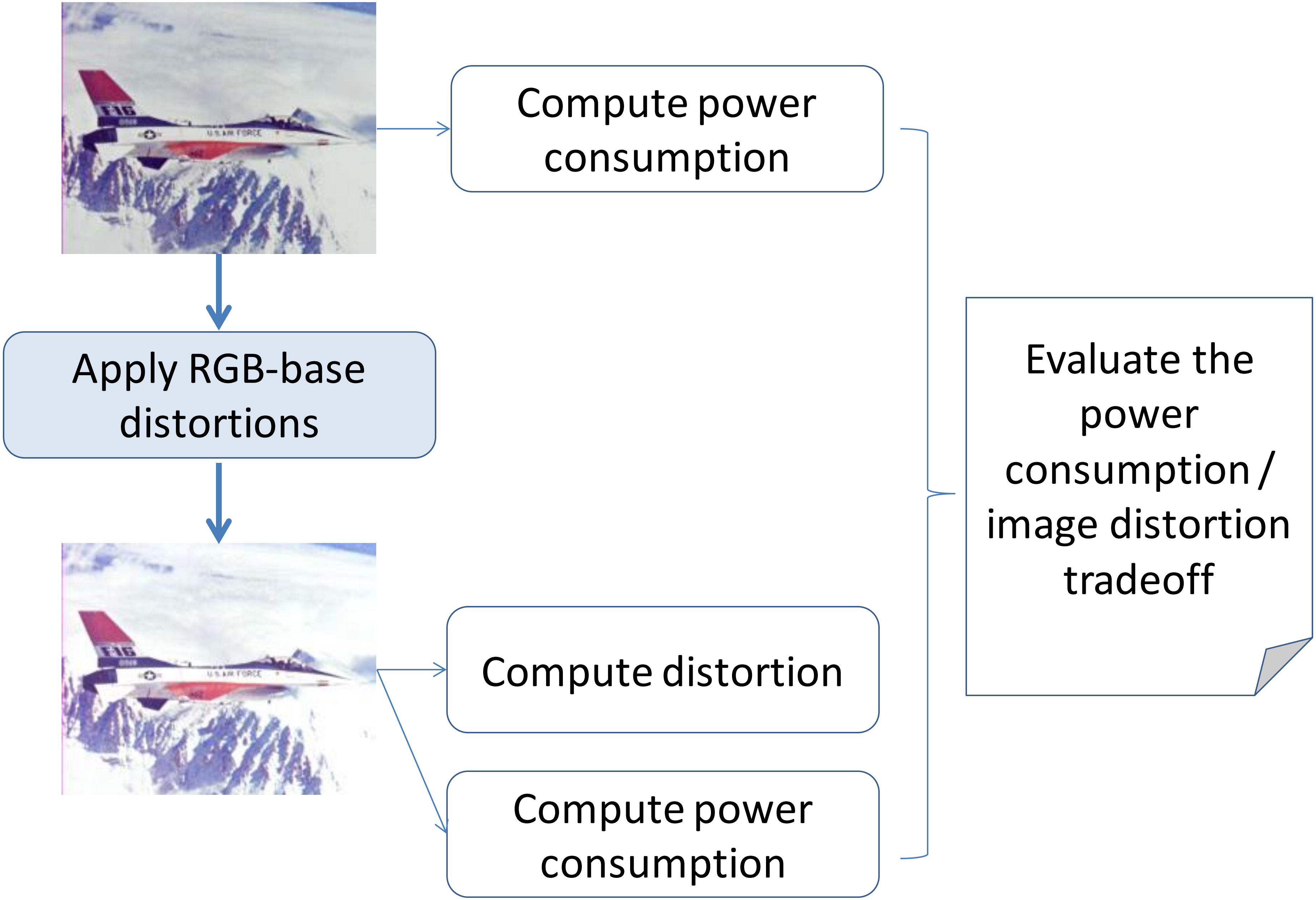
We normalize our value obtained of the distortion to make

4. Evaluation of image distortion (cont'd)

- Alternative: Mean Structural Similarity Index (MSSIM).
  - More complex metric involving measures of brightness, contrast and structural distortion
  - Implementation provided by MATLAB: **ssim(A, B)**
  - Score between 0 and 1 (1 = identical images).
    - Convert to % simply by multiplying * 100

- **TASK:** Compare the two distortion metrics and determine which one correlates better with visual differences

For every transformation we do, compare the ditortion with the two metrics and compare the two methods: which is the one that re


Compute power consumption
Apply RGB-base distortions
Evaluate the power consumption / image distortion tradeoff
Compute distortion
Compute power consumption

5. Evaluation of various strategies on image modification

- **TASK:** Experiment image manipulation strategies to reduce power consumption:
  - Pixel-wise transformations
    - Work on colors
  - Histogram equalization
    - Work on luminance
  - Other types of brightness/contrast modifications
- Apply to all images

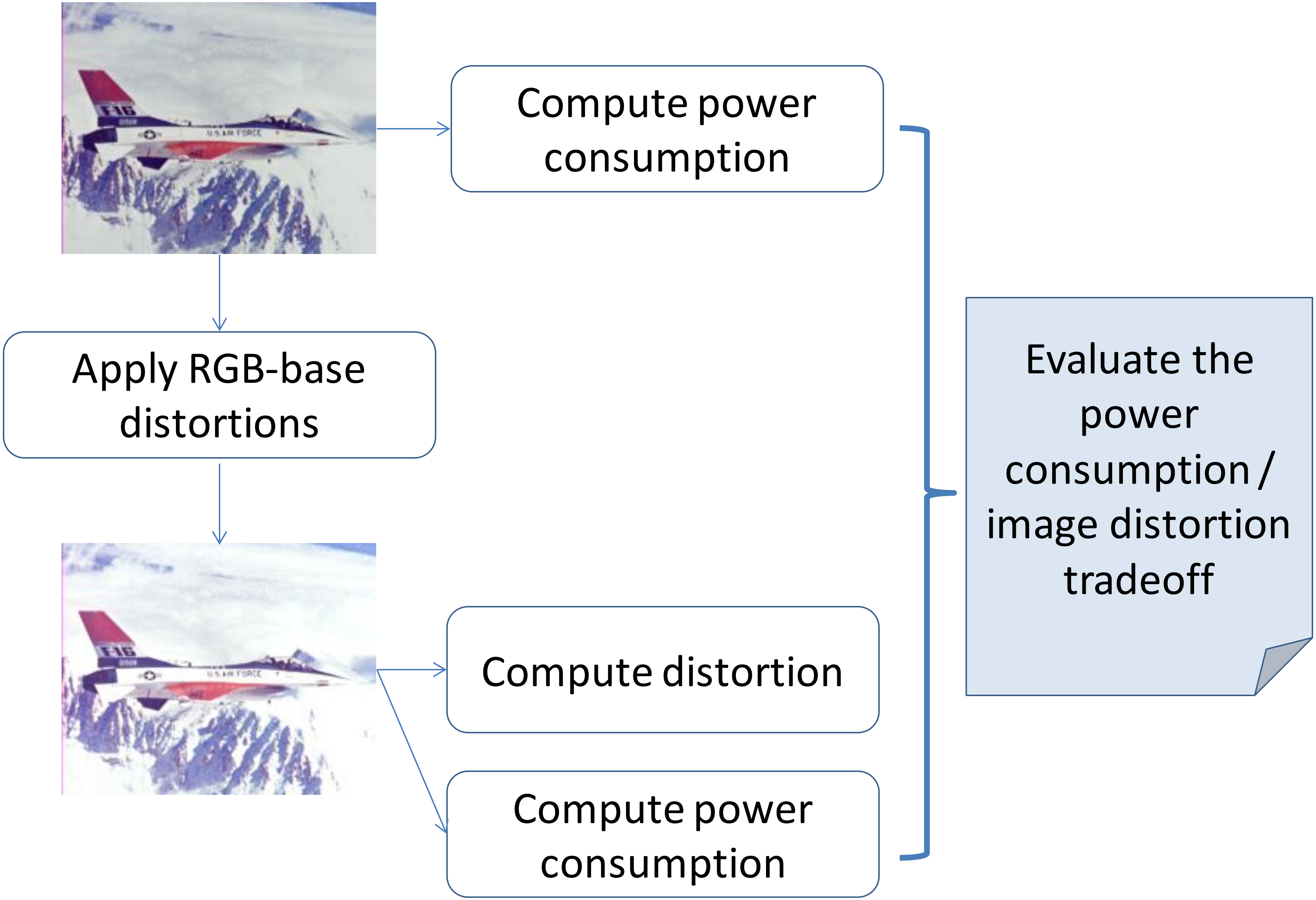
Polynomial transformation between the input pixel and the output pixel (cubic p

5. Evaluation of various strategies on image modification

- **Hungry-blue:**
  - Blue is the more costly component..
  - So, subtract a constant value to the blue component
  - Will introduce a distortion but reduces power
    - Depending on the weight of the blue component

- **Histogram equalization:**
  - Flatten the histogram of the image, e.g., by working on luminance…
  - Requires to work in the HSV color space
    - Hue – saturation – value (i.e., luminance)
  - Introduces distortion

- **Other types of brightness/contrast transformations:**
  - E.g. Convert to HSV and scale the value component (V → k*V with k < 1) or do some more complex transformation
  - use your imagination


Compute power consumption
Apply RGB-base distortions
Compute distortion
Compute power consumption
Evaluate the power consumption / image distortion tradeoff

6. Analysis

- Analyse power/distortion tradeoff
  - Do different images behave differently?
  - What changes in terms of power consumption with different manipulation strategies?
  - How can I save more power with lower distorsions?
  - …

Differences between images ar

- **TASK:** Compare the transformations you applied and find the solution that **minimizes** power consumption under a **maximum average distortion** constraint (1%, 5%, 10%)
  - Use both distortion metrics

Implement a script

6. Analysis

- Example:
  - Blue reduction
    - Power saving 29.11%
    - Distortion 30.99%
  - Histogram equalization
    - Power saving 11.99%
    - Distortion 8.46%

Original image

After blue reduction

After histogram equalization

How to:

1. Learn how to load an image in MATLAB
   – And how to convert it to RGB

2. Write the function to estimate power consumption

3. Write the function to estimate distortion