Video Analysis

LV 188.329, WS 2016/17

Video recording – spatial color interpolation

Ao. Univ.-Prof. Margrit Gelautz

margrit.gelautz@tuwien.ac.at

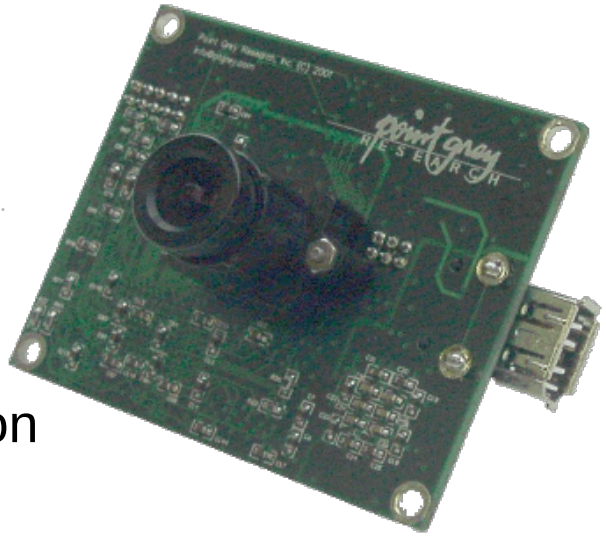
FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Video recording – spatial color interpolation

- Background
  - To save costs, digital cameras often make use of only one image sensor (*single detector*).
- Principle
  - By using a *Color Filter Array (CFA)* between lens and sensor, color information is obtained.
  - This way an intensity value of a single color component is assigned to each pixel according to a fixed scheme.
  - Methods to calculate the remaining 2 color components at each pixel position are required to reconstruct the original color images (*spatial color interpolation* or *color demosaicking)*.

Example of a camera



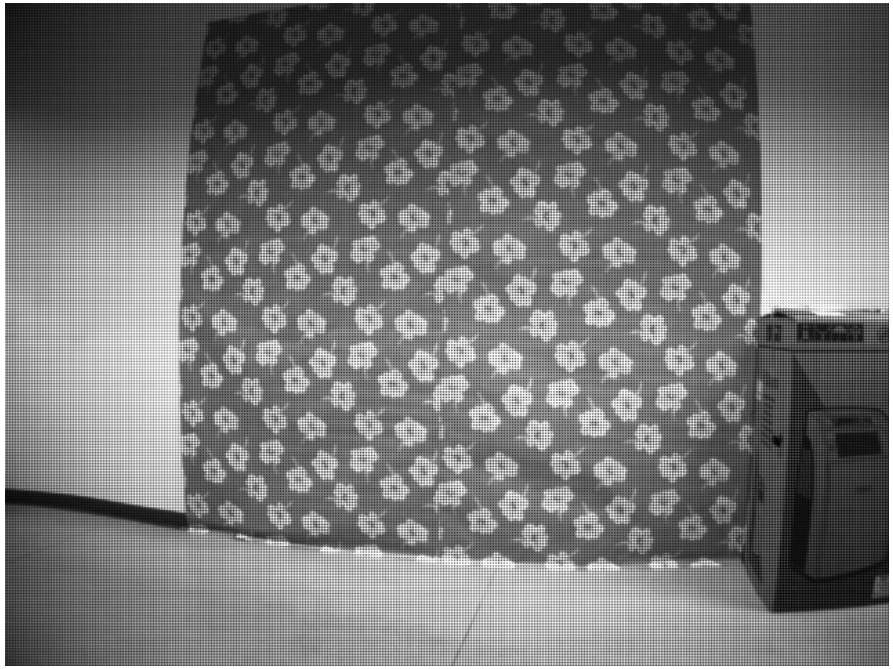
- Dragonfly (Manufacturer: Pointgrey)
- Specifications
  - 6-pin IEEE-1394 interface
  - Multiple frame rates
  - Automatic camera-to-camera synchronization
  - **640x480 or 1024x768 24-bit true color or 8-bit gray scale images**
  - …
- Supported Formats (Excerpt of the data sheet)
  - **B&W models: 8-bit Mono**
  - **Color models: 8-bit Bayer tiled image (color space conversion done on the host computer)**

Bayer Pattern

- There are many different Color Filter Array schemes.
- The Bayer Pattern is of widespread use - it employs the 3 primary colors red, green and blue as filter elements.
- The corresponding layout of the color filter elements is shown in the figure on the right.

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

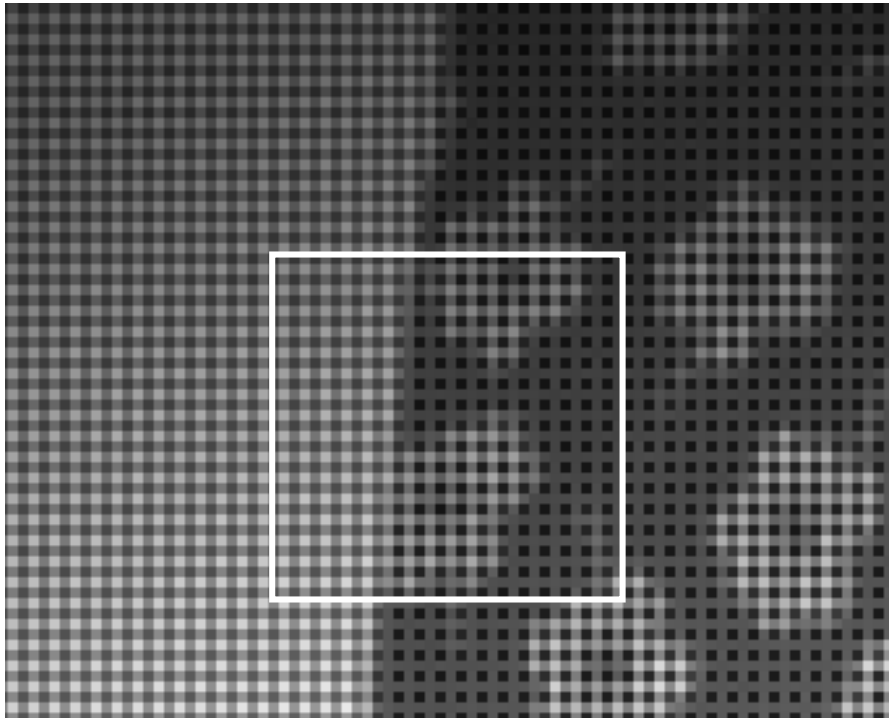
Comparison: original and calculated image



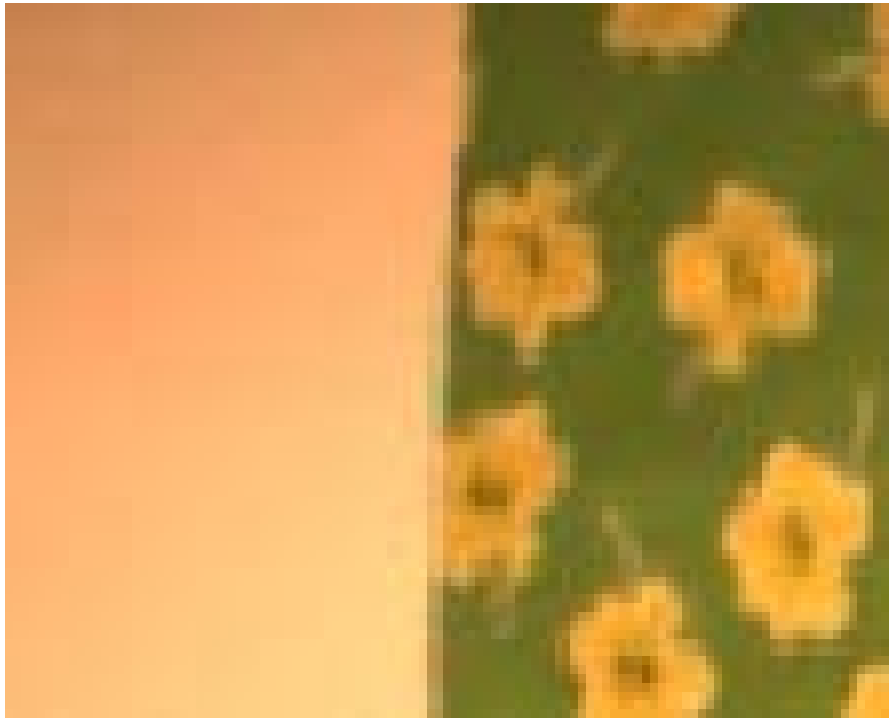
Intensity/grey scale image taken by the image sensor *(stippled image)*.

Calculated (reconstructed) color image.

Detail 1

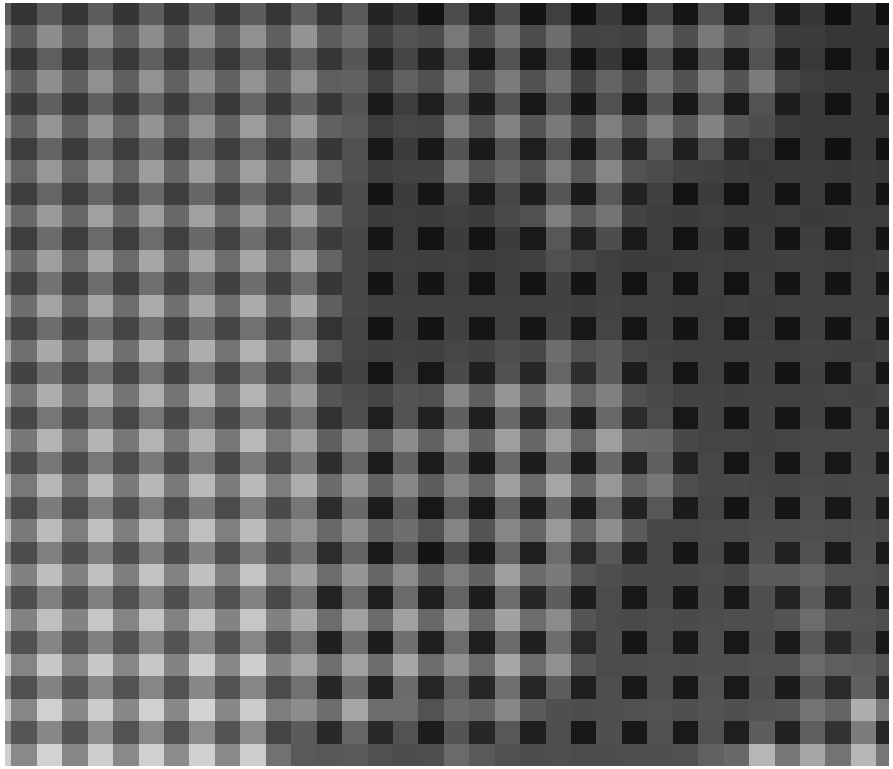


Stippled image (detail)



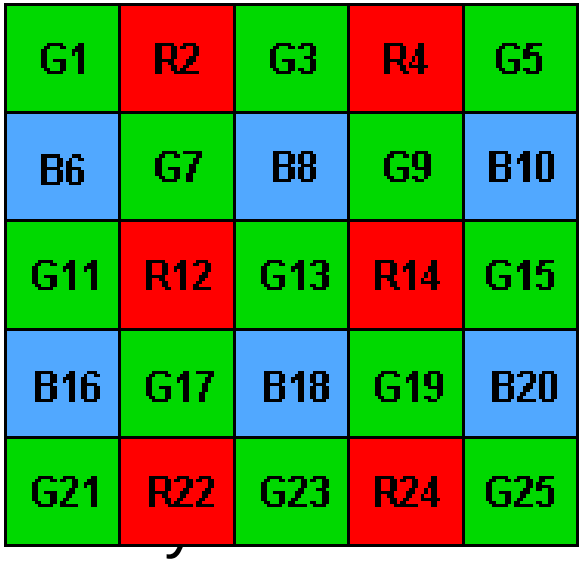
Color image (detail)

Detail 2



Zoom

(see selection on previous slide)

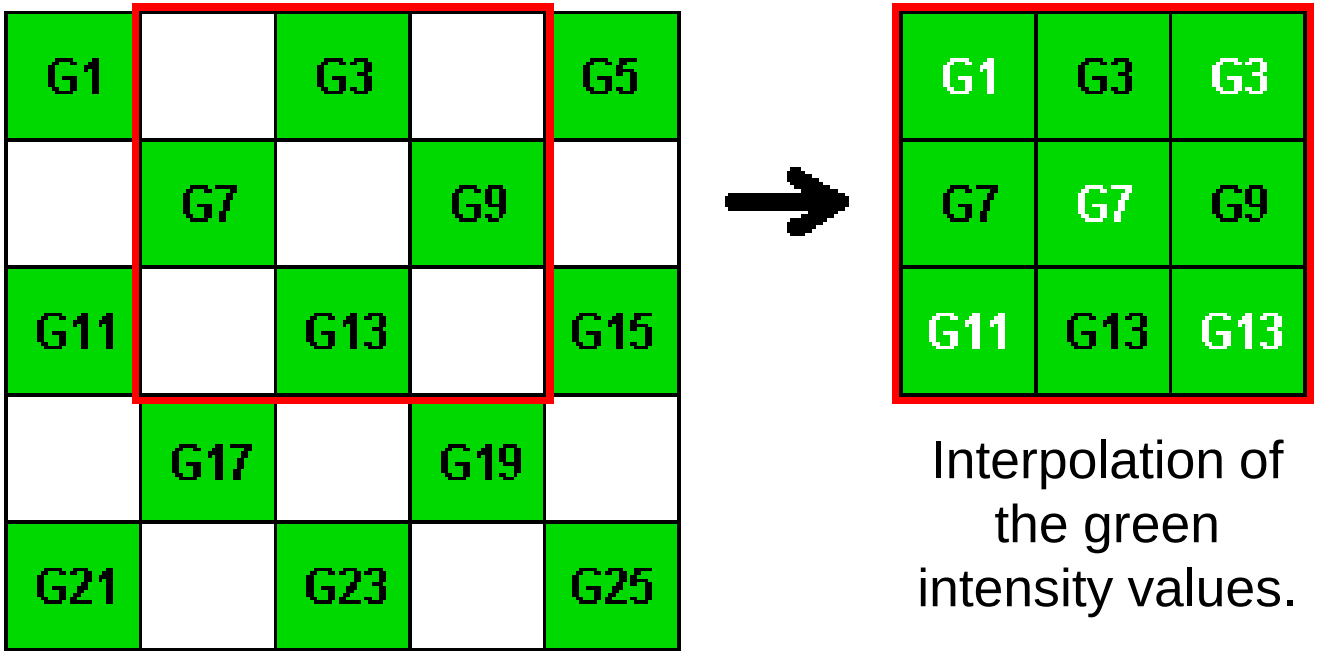


Question: Can you tell the location of the green pixels in the left image?

Algorithms for spatial color interpolation

- Two types of algorithms for spatial color interpolation:
  1. **Non-adaptive algorithms** perform the interpolation for each pixel according to a pre-defined scheme.
     - Nearest Neighbor Replication
     - Bilinear Interpolation
     - …
  2. **Adaptive algorithms** take into account the local properties of the image (pixel neighborhood).
     - e.g., incorporation of edge information

Method 1: Nearest neighbor assignment



Interpolation of the green intensity values.

Assignment using the *Nearest Neighbor* principle: In this example, we always use the left intensity value. (Instead, we could also use the upper, lower or right value.)

Method 2: Bilinear interpolation

- Interpolation of green pixels
  - The average value of the upper, lower, left and right neighbors yields the interpolated value of the pixel under consideration:

    G8 = (G3 + G7 + G9 + G13) / 4

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Method 2: Bilinear interpolation

- Interpolation of red/blue pixels
  - At a pixel position that contains a **green value**: Average of neighboring pixels of the respective color:

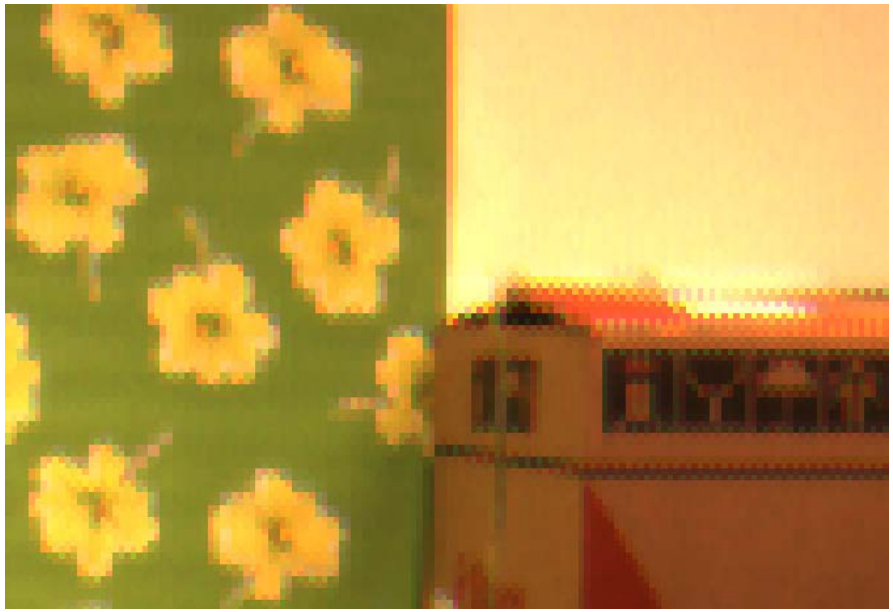
    B9 = (B8 + B10) / 2
    R9 = (R4 + R14) / 2

  - At a pixel position that contains a **red or blue value**: Average of diagonally neighboring 4 pixels:

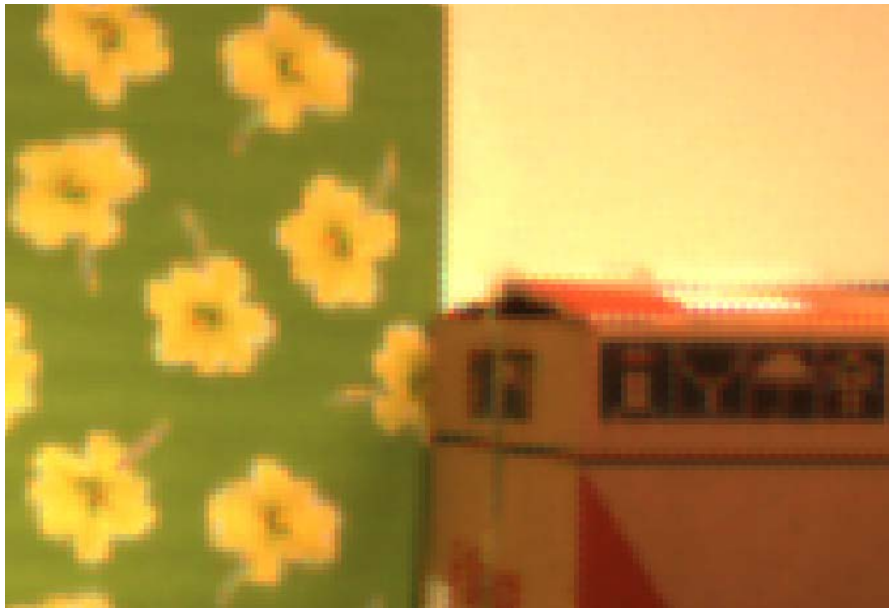
    R8 = (R2 + R4 + R12 + R14) / 4
    B12 = (B6 + B8 + B16 + B18) / 4

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

Results of the interpolation



(a) Nearest Neighbor



(b) Bilinear Interpolation

Advantages: Low processing costs with both methods, especially with (a).

Problems: Block generation (a), blurry image (b), disturbing artifacts at the edges with (a) and (b).

Artifacts at the edges

(a) The so-called *zipper effect* at the edges (e.g., caused by Bilinear Interpolation)

(b) Smoother color gradients can be achieved by applying more sophisticated interpolation methods.

Method 3: Edge Sensing Interpolation - Principle

- This method is an example of an **adaptive algorithm**: The properties of adjacent pixels are considered, and influence the decision which pixels of the neighborhood are used to calculate the interpolated value.

- **Gradients** are computed in order to take into account **edge information**.

- In the following, we will describe the interpolation of green pixels based on gradient information.

Method 3: Edge Sensing Interpolation algorithm

- Interpolation of green pixels:
  - Definition of horizontal and vertical gradient at every red and blue position.

    e.g., for *B8*:
    $\Delta H = |G7 - G9|$ and $\Delta V = |G3 - G13|$
  - Definition of a threshold value *T*
  - Calculation of the interpolated values:

G1	R2	G3	R4	G5
B6	G7	B8	G9	B10
G11	R12	G13	R14	G15
B16	G17	B18	G19	B20
G21	R22	G23	R24	G25

IF $\Delta H < T$ ***AND*** $\Delta V > T$
    $G8 = (G7 + G9) / 2;$
ELSE IF $\Delta H > T$ ***AND*** $\Delta V < T$
    $G8 = (G3 + G13) / 2;$
ELSE
    $G8 = (G3 + G7 + G9 + G13) / 4;$

Methode 3: Edge Sensing Interpolation - Result

(a) For comparison, the results of *Bilinear Interpolation* are shown above.

(b) Reconstruction using *Edge Sensing Interpolation*: One can recognize a better (sharper) reconstruction of the edges than in (a).