

PjE – Projet Encadré IVI

Semaine 2 : acquisition vidéo

Master ASE : <http://master-ase.univ-lille1.fr/>
Master Informatique : <http://www.fil.univ-lille1.fr/>
Spécialité IVI : <http://master-ivi.univ-lille1.fr/>

Plan du cours

1 – Échantillonnage et quantification

échantillonnage temporel : cadence, temps d'intégration

échantillonnage spatial : grille, résolution, taille de l'image

échantillonnage spectral : images monochromes ou couleur

quantification : du continu au numérique

2 – Caméra couleur IDS uEye 1225-C et son capteur

caractéristiques de la caméra et du capteur

matrice CMOS et matrice de filtres couleur (Bayer)

réponses spectrales : utilisation en couleur ou en infrarouge

3 – Chaîne d'acquisition

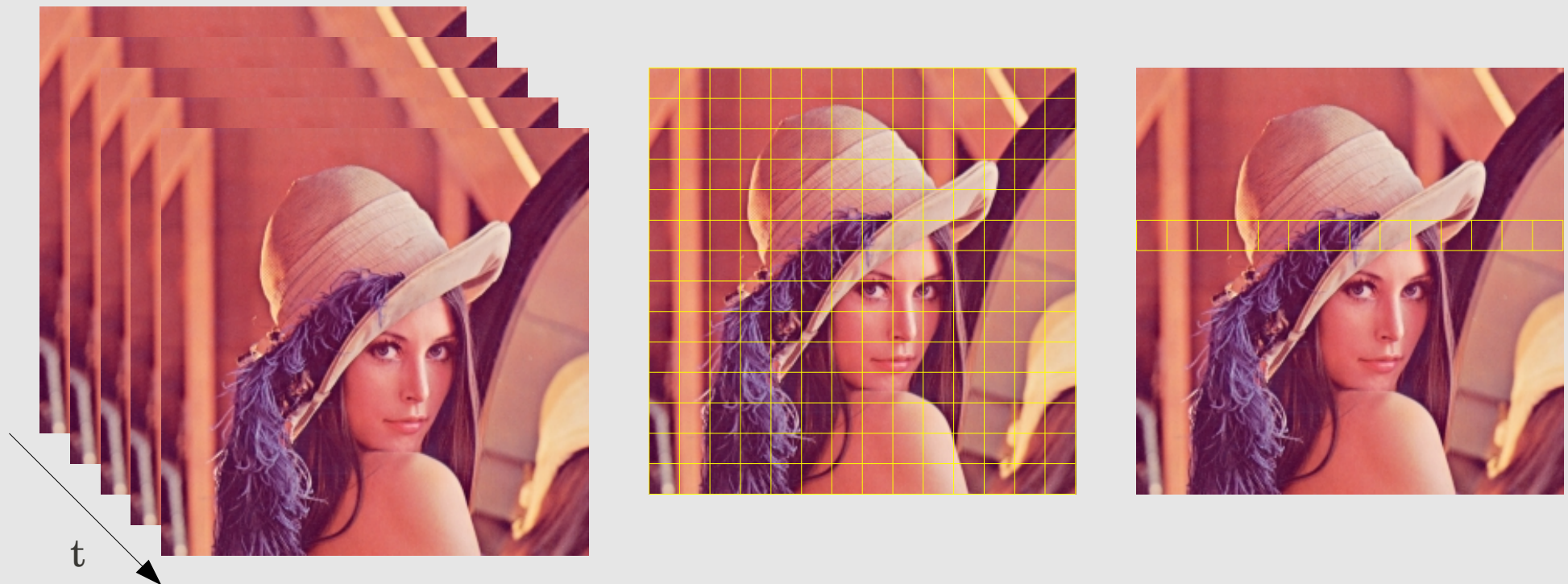
éléments de la chaîne : caméra, bus USB, pilote logiciel

rôle du pilote logiciel (driver)

stockage des images en mémoire, mode d'acquisition

Du monde réel à l'image numérique

Transformation de l'espace 3D en un tableau de valeurs
formation de l'image (projection) à travers un système optique
échantillonnage temporel, spatial et spectral par le **capteur**
numérisation par le capteur et / ou le système de traitement



Échantillonnage temporel

Cadence d'acquisition des images

on ne peut pas mesurer l'information lumineuse de façon **continue** dans le temps, donc on prend des **échantillons**.
de ce fait, on mesure **périodiquement** l'information lumineuse.
la **cadence** d'acquisition est l'**inverse** de la **période** d'acquisition.

Exemple

cadence vidéo en France : 25 images.s^{-1} , période de 40 ms.

image 1

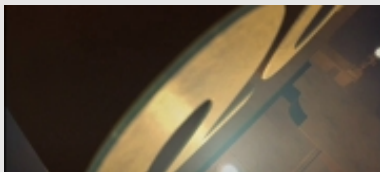


image 2

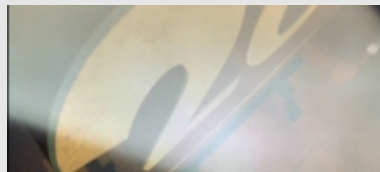


image 3

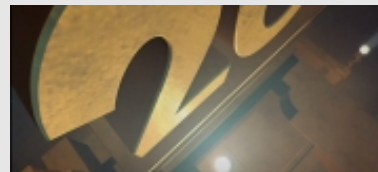
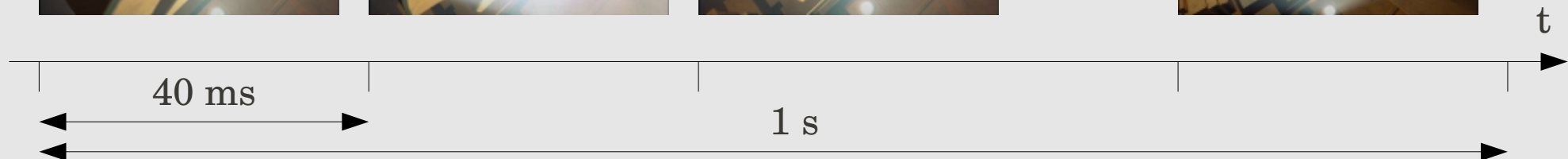
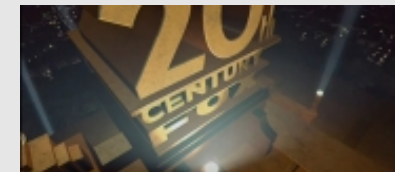


image 25



Intégration de l'information lumineuse

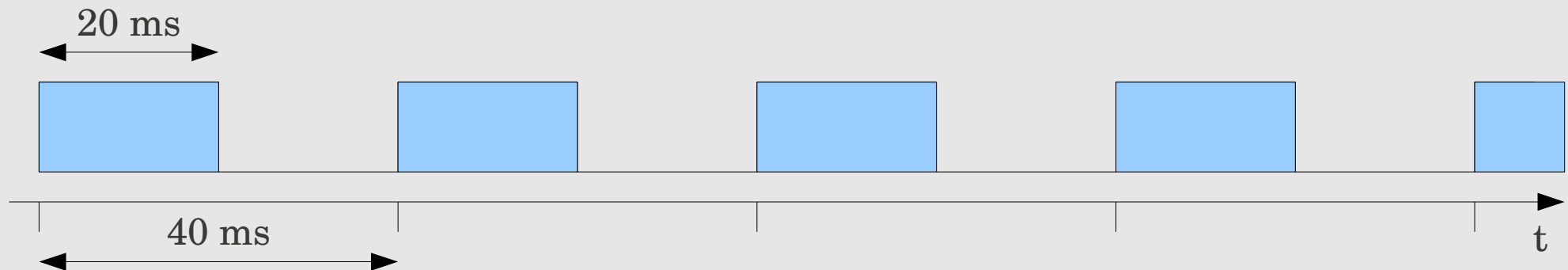
Temps d'intégration

un capteur élémentaire devrait théoriquement mesurer **instantanément** le **flux énergétique** qu'il reçoit, c'est à dire l'énergie par unité de temps ($W = J.s^{-1}$).

en pratique, un capteur mesure l'**énergie totale** (en J) qu'il reçoit pendant un **intervalle de temps** (en s) de durée non nulle, appelé **temps d'intégration**.

Exemple

cadence d'acquisition = 25 im.s⁻¹, temps d'intégration = 20 ms

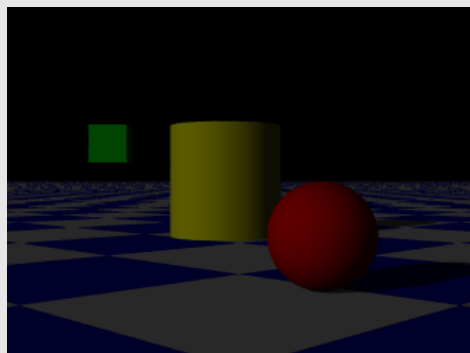


Compromis sensibilité / flou de bougé

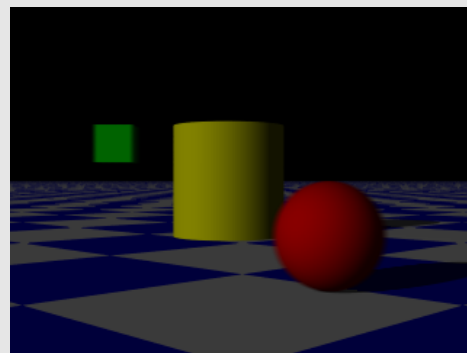
Pourquoi ?

en **augmentant** le temps d'intégration, on améliore la **sensibilité**.
si un objet **bouge**, son image bouge également sur le capteur.
dans ce cas, le capteur accumule des mesures **non constantes**, ce
qui entraîne un phénomène appelé **flou de bougé**.
un **point mobile** apparaît comme une **ligne** dans l'image.

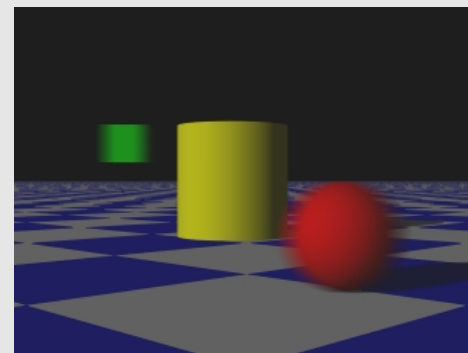
Exemple



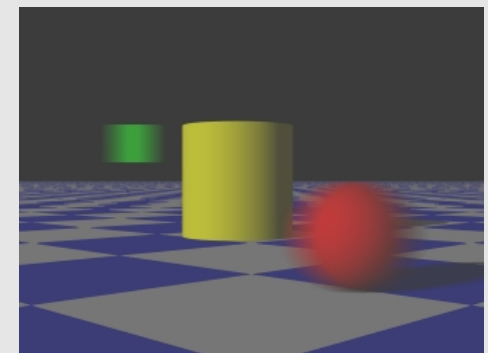
$t = 1 \text{ ms}$



$t = 10 \text{ ms}$



$t = 20 \text{ ms}$



$t = 30 \text{ ms}$

Image continue : fonction de deux variables

Fonction image

fonction $I : \mathbb{R}^2 \rightarrow \mathbb{R}^n, (x,y) \rightarrow I(x,y)$

n est le nombre de **composantes** de l'image ($n \geq 1$)

dans ce cas, l'espace \mathbb{R}^2 est appelé **plan image**.

Support de l'image

sous ensemble de \mathbb{R}^2 de définition de $I(x,y)$, de surface **finie**, en général un **rectangle**, de côtés T_x et T_y .

Exemple à 1 composante :

$I(x,y) = e^{-(x^2+y^2)} \cdot \cos(x) \cdot \sin(y)$,
pour $x \in [-\pi, \pi]$ et $y \in [-\pi, \pi]$
(support carré)

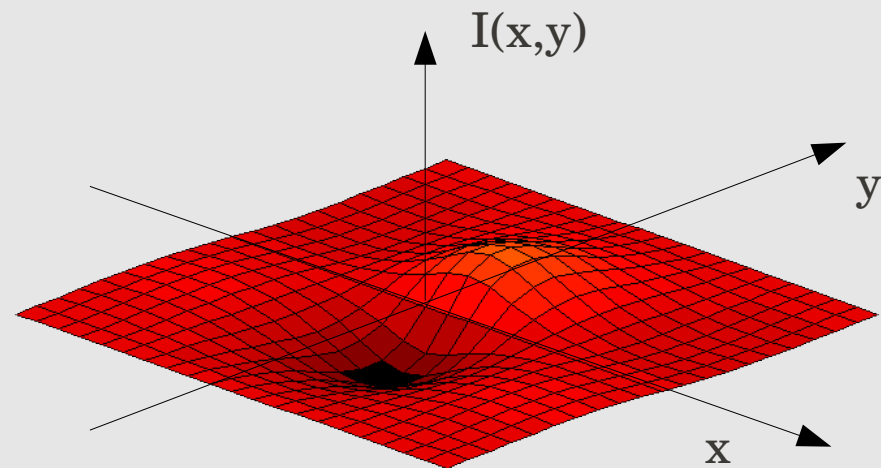


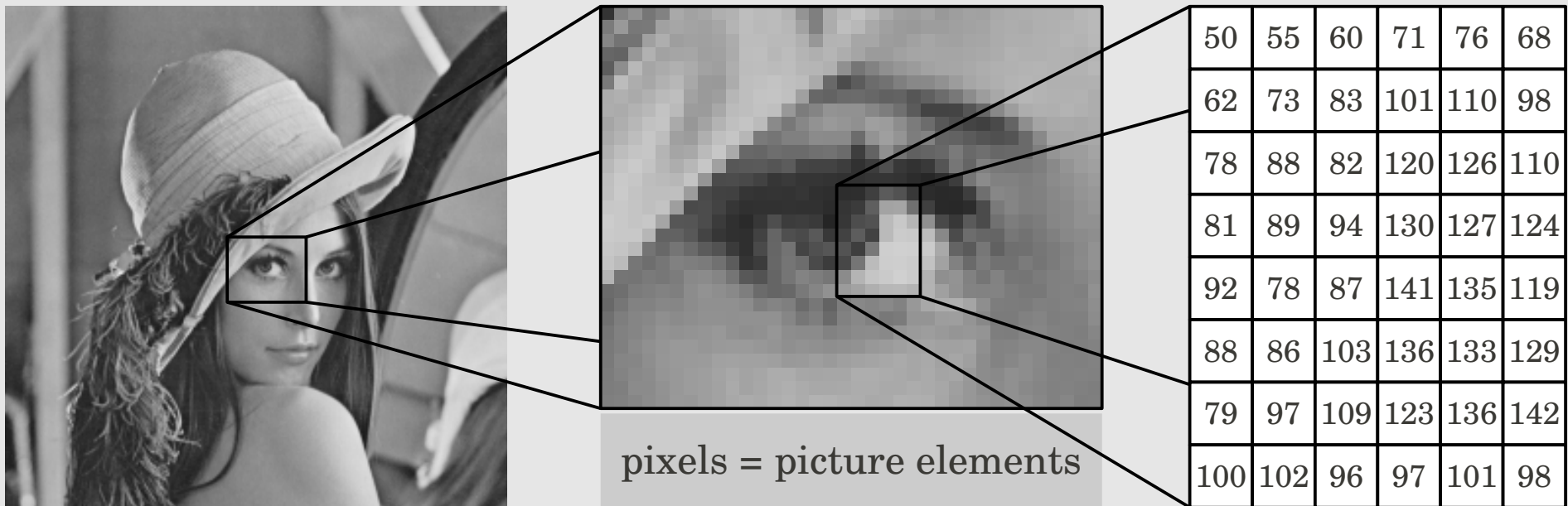
Image numérique : tableau bidimensionnel de valeurs

Échantillonnage de la fonction image

échantillonnage en x et en y de la fonction, ne conserve que les valeurs pour les points $(x,y) = (c.\Delta x, l.\Delta y)$ avec c, l entiers

tableau de valeurs $I : \mathbb{Z}^2 \rightarrow \mathbb{R}^n, (c,l) \rightarrow I(c.\Delta x, l.\Delta y)$

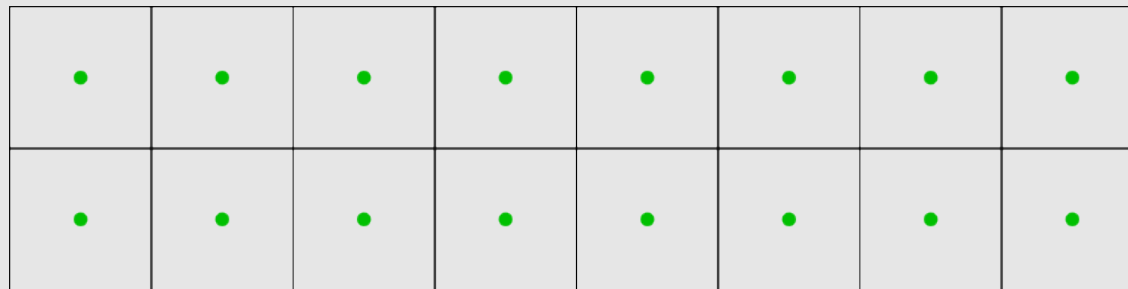
Exemple à 1 composante :



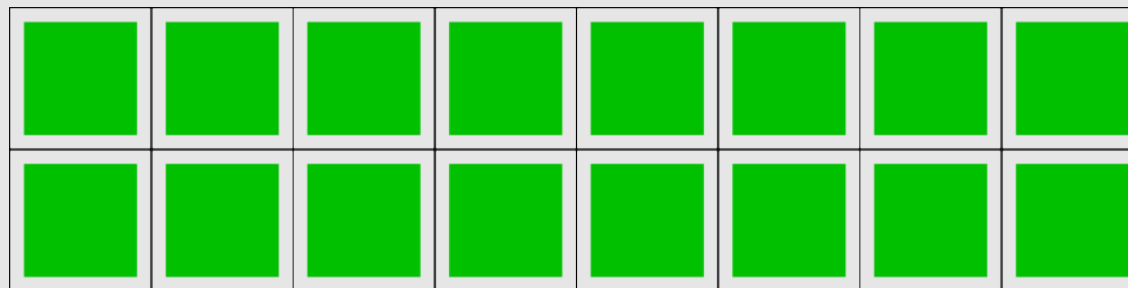
Échantillonnage spatial et résolution (1/3)

Type d'échantillonnage spatial

parfait : la valeur d'un pixel est une mesure du **flux énergétique** reçu par un **point** sensible (de surface nulle) du capteur.



réel : la valeur d'un pixel est une mesure du flux énergétique reçu par une **surface** sensible élémentaire. C'est le **produit** de l'**éclairement énergétique** par la **surface** de l'élément sensible.



Échantillonnage spatial et résolution (2/3)

Résolution d'une image

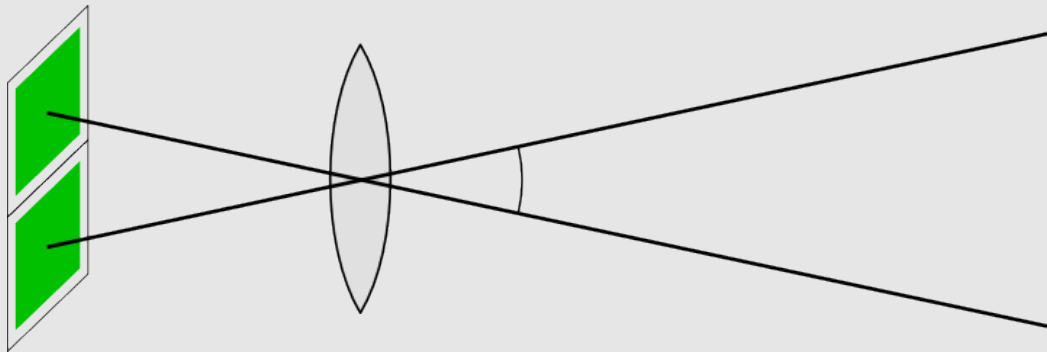
nombre de pixels par **unité de longueur**. La résolution s'exprime en pixels.m^{-1} (ou en pixels.in^{-1} dans les pays anglo-saxons).

la résolution est l'**inverse** du pas de la grille d'échantillonnage.

la résolution est le **rapport** du nombre de pixels divisé par la **dimension** du capteur.

Pouvoir de résolution

mesure, par un **angle**, la capacité d'un système d'acquisition d'image (optique + capteur) à **distinguer** des détails fins.



oeil = 1/60 de degré

télescope = 1/30000 de degré

Échantillonnage spatial et résolution (3/3)



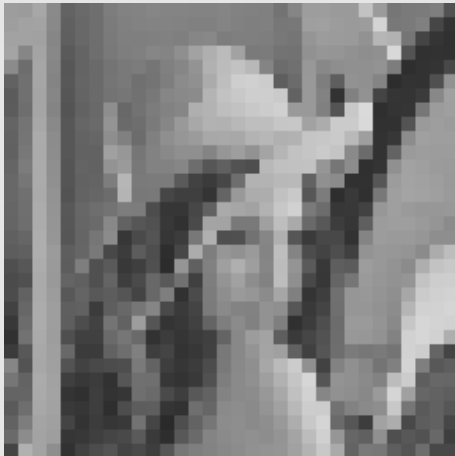
256 x 256



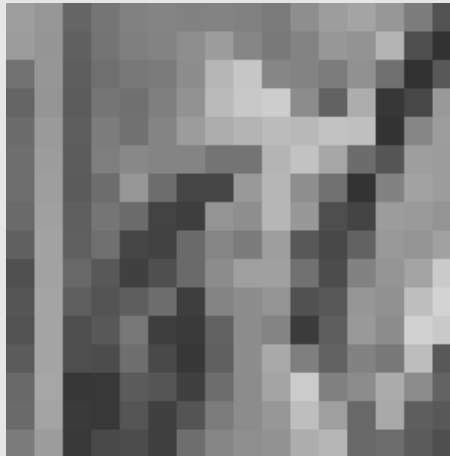
128 x 128



64 x 64



32 x 32



16 x 16



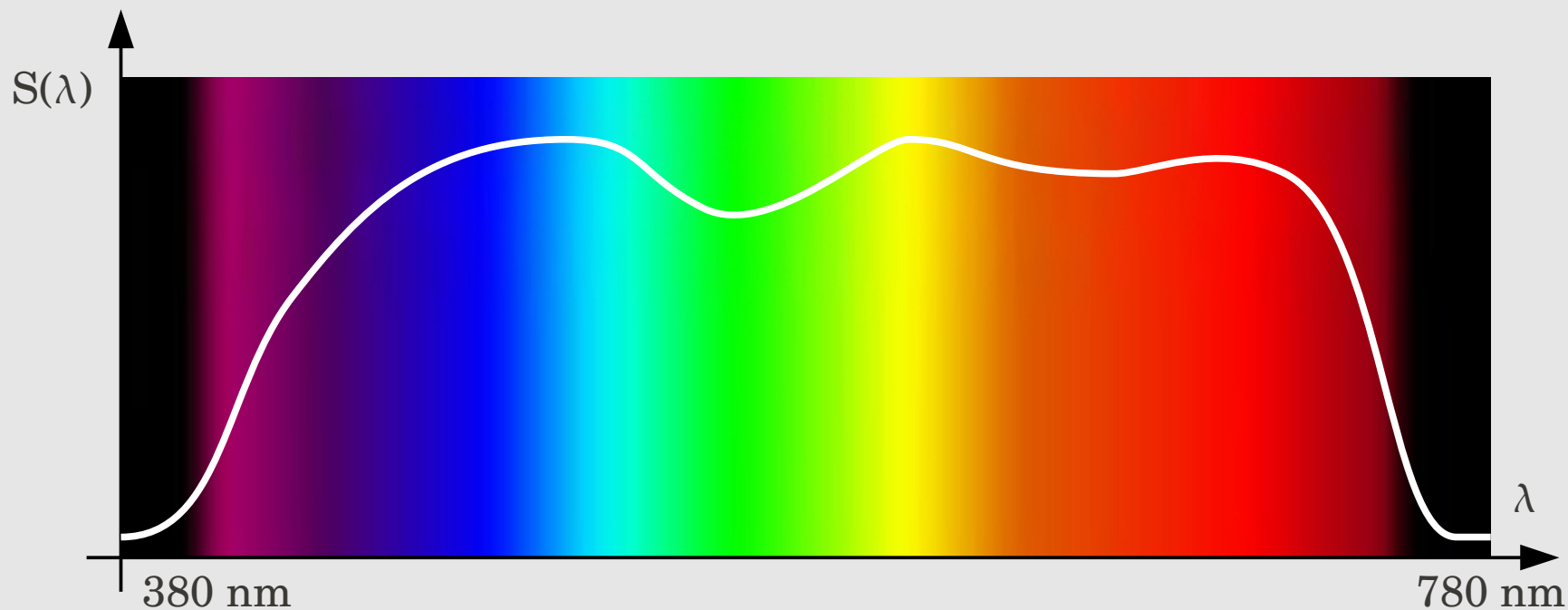
8 x 8

Échantillonnage spectral : image monochrome (1/2)

Image monochrome : 1 échantillon

pour chaque pixel on mesure le **flux énergétique total**, c'est à dire pour **toutes** les longueurs d'onde du spectre.

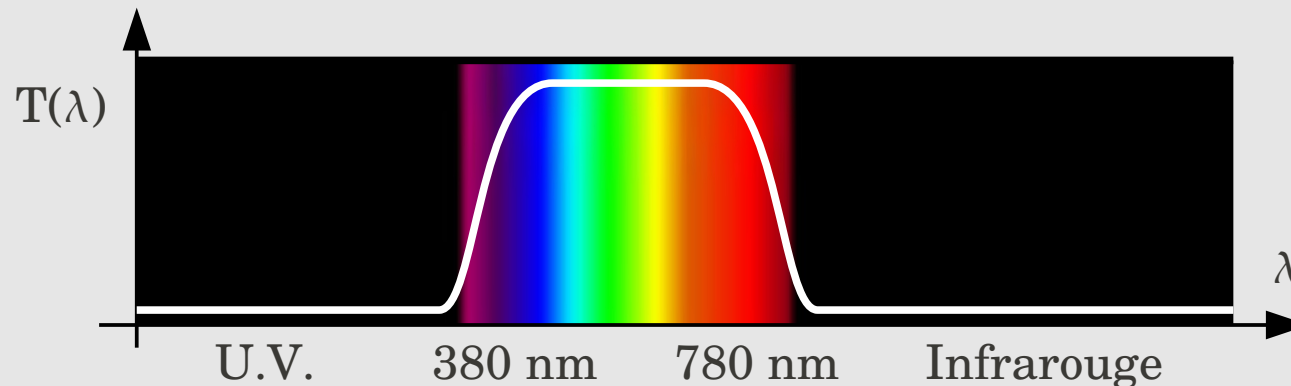
en pratique, un capteur monochromatique est caractérisé par une courbe de **sensibilité spectrale**.



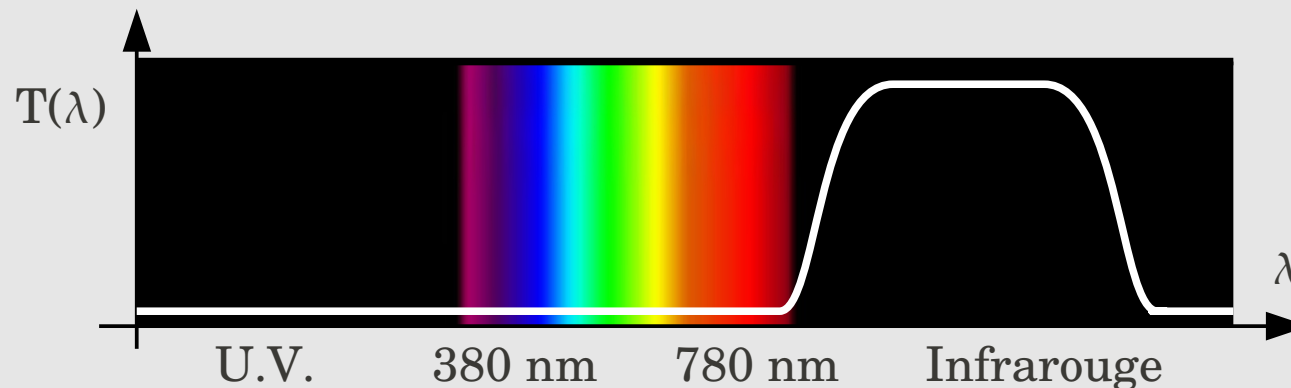
Échantillonnage spectral : image monochrome (2/2)

Insertion d'un filtre devant le capteur

les longueurs d'ondes de la lumière **traversant** le filtre sont plus ou moins **absorbées** et **transmises**. Le filtre est caractérisé par sa **courbe de transmission spectrale**.



filtre visible :
image monochrome
standard (N&B)

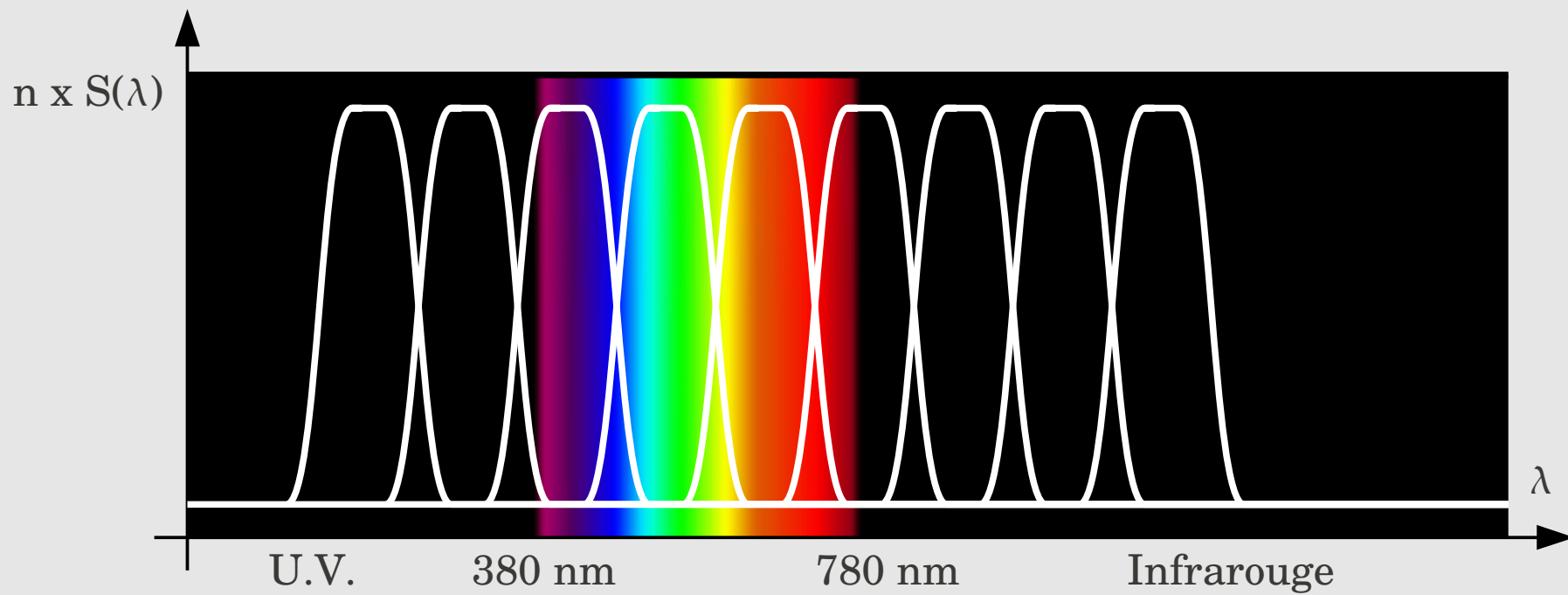


filtre infrarouge :
proche ou lointain
(thermique)

Échantillonnage spectral : image hyperspectrale (1/2)

Échantillonnage du spectre en n bandes

pour acquérir une **composante**, on utilise un capteur qui a une sensibilité spectrale correspondant à une **bande du spectre**.
une composante est mesurée par l'intégrale du **flux énergétique monochromatique**, pondérée par la sensibilité spectrale.

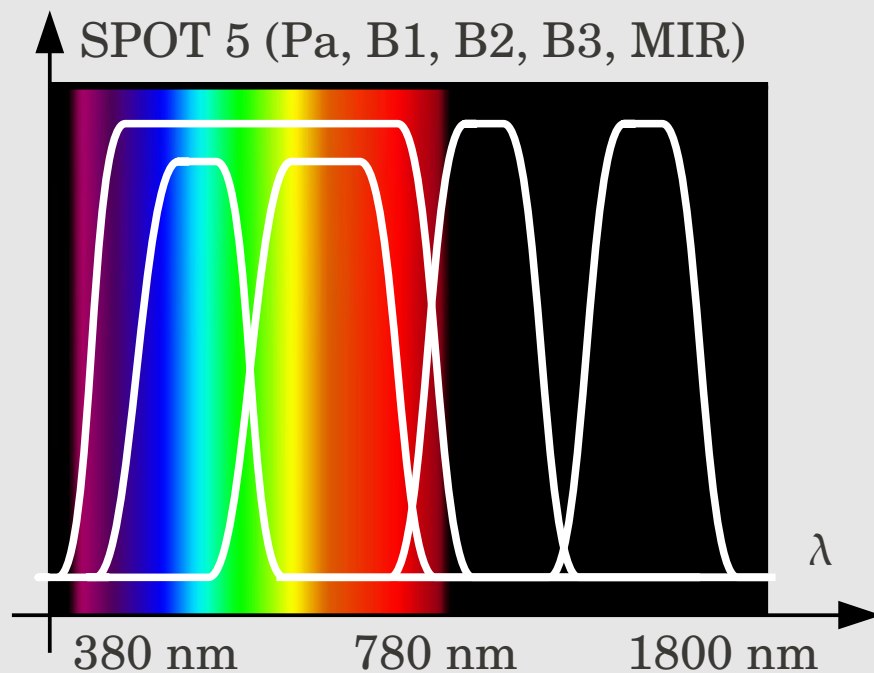


Échantillonnage spectral : image hyperspectrale (2/2)

Image multi-spectrale

les bandes sont en **nombre limité** et sont choisies parce que leurs longueurs d'ondes ont des propriétés particulières.

exemple : **imagerie satellitaire**, la surface de la terre est observée dans le visible et **l'infrarouge** (végétation)

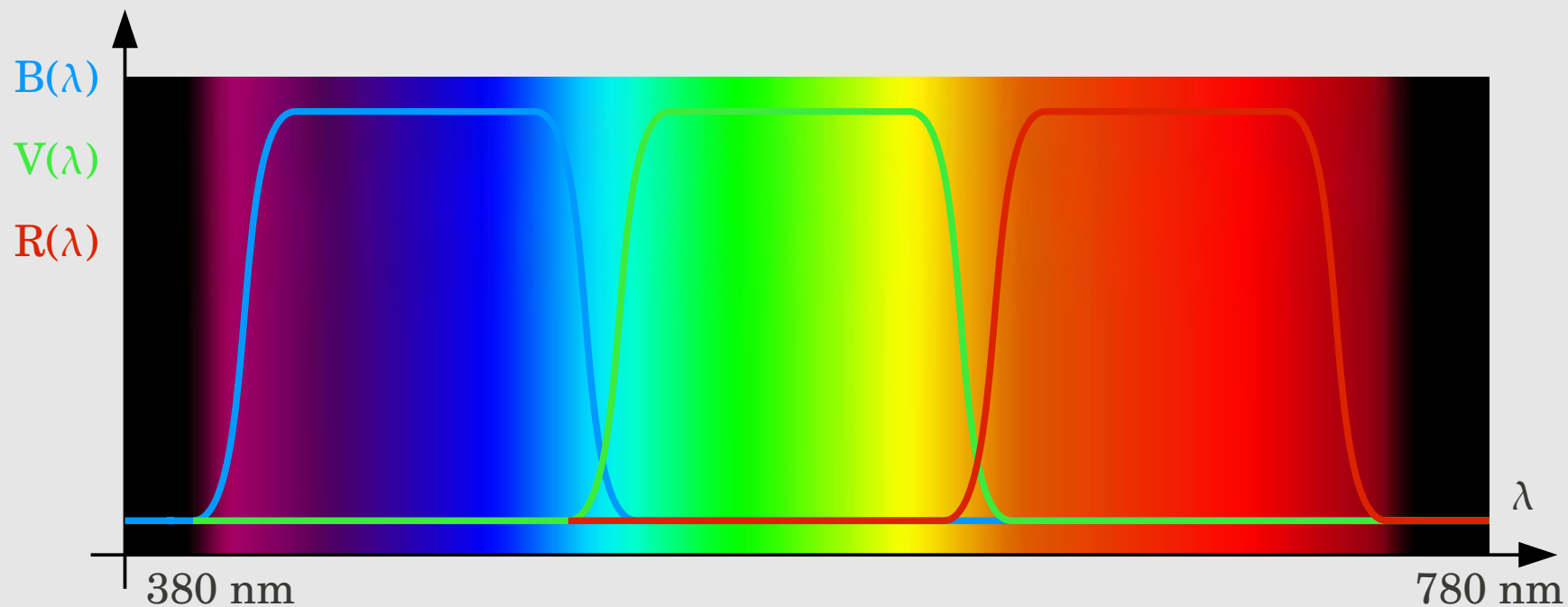


Échantillonnage spectral : image couleur

Image couleur, 3 composantes : Rouge, Verte, Bleue

on échantillonne le **spectre visible** dans trois bandes.

les trois bandes (rouge, verte, bleue) ont été sélectionnées afin de correspondre à la **vision humaine standard**.



Numérisation des composantes : quantification

Valeurs numériques

chaque valeur d'une composante de la fonction image est représentée par un **mot binaire**, codé sur un nombre fini de bits.
pour un mot de m bits, la valeur varie entre 0 et $2^m - 1$.
exemple : sur 8 bits, composantes entre 0 et 255.

Taille des données image

dimensions du support : N_x pixels sur N_y lignes.
nombre de composantes : n , nombre de bits de quantification : m
→ taille en bits = $N_x \cdot N_y \cdot n \cdot m$

Exemples:

image monochrome binaire 256x256 : 65.536 bits = 8 Ko

image spot 2048x2048, 4 canaux, 12 bits : 201.326.592 bits = 24Mo

Effet de la quantification sur le rendu visuel



256 niveaux



64 niveaux



16 niveaux



8 niveaux



4 niveaux



2 niveaux

Caméra numérique USB, IDS uEye 1225 LE-C

Caractéristiques

- caméra numérique couleur
- capteur **mono-CMOS**, filtre CFA **Bayer**
- taille maximale de l'image 752x480 pixels
- réglage de la **zone d'intêret** (AOI)
- **binning** vertical et/ou horizontal
- sortie USB 2.0
- **monture CS**, objectifs à visser



Fabriquant

IDS Imaging : <http://www.ids-imaging.de/>

Distributeur français

STEMMER Imaging : <http://www.stemmer-imaging.fr/>

Capteur, Micron MT9V022

Caractéristiques

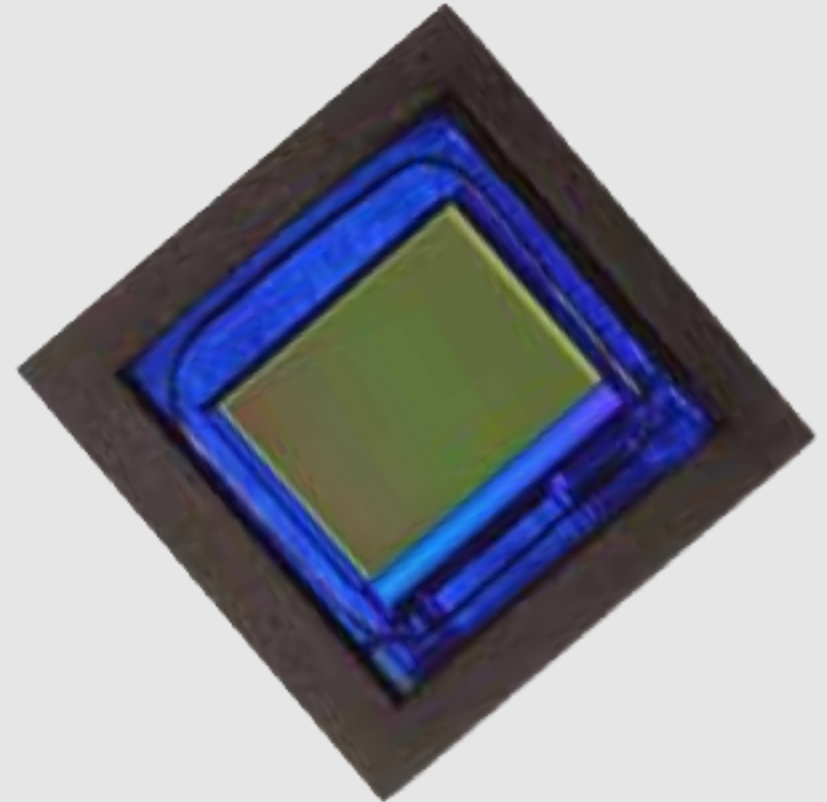
- format optique **1/3 de pouce**
- capteur 4.51mm(H) x 2.88mm(V)
- 752x480 pixels **utiles**
- pixels de **6 x 6 microns**

Cadence et exposition

- fréquence pixel de **2 à 40 MHz**
- temps d'intégration: **80 μ s à 5.5 s**
- jusqu'à **87 im/s** à pleine résolution

3 types de pixels, filtre CFA

- 90240 pixels sensibles au **rouge + infrarouge**
- 180480 (x2) pixels sensibles au **vert + infrarouge**
- 90240 pixels sensibles au **bleu + infrarouge**



Organisation spatiale des pixels

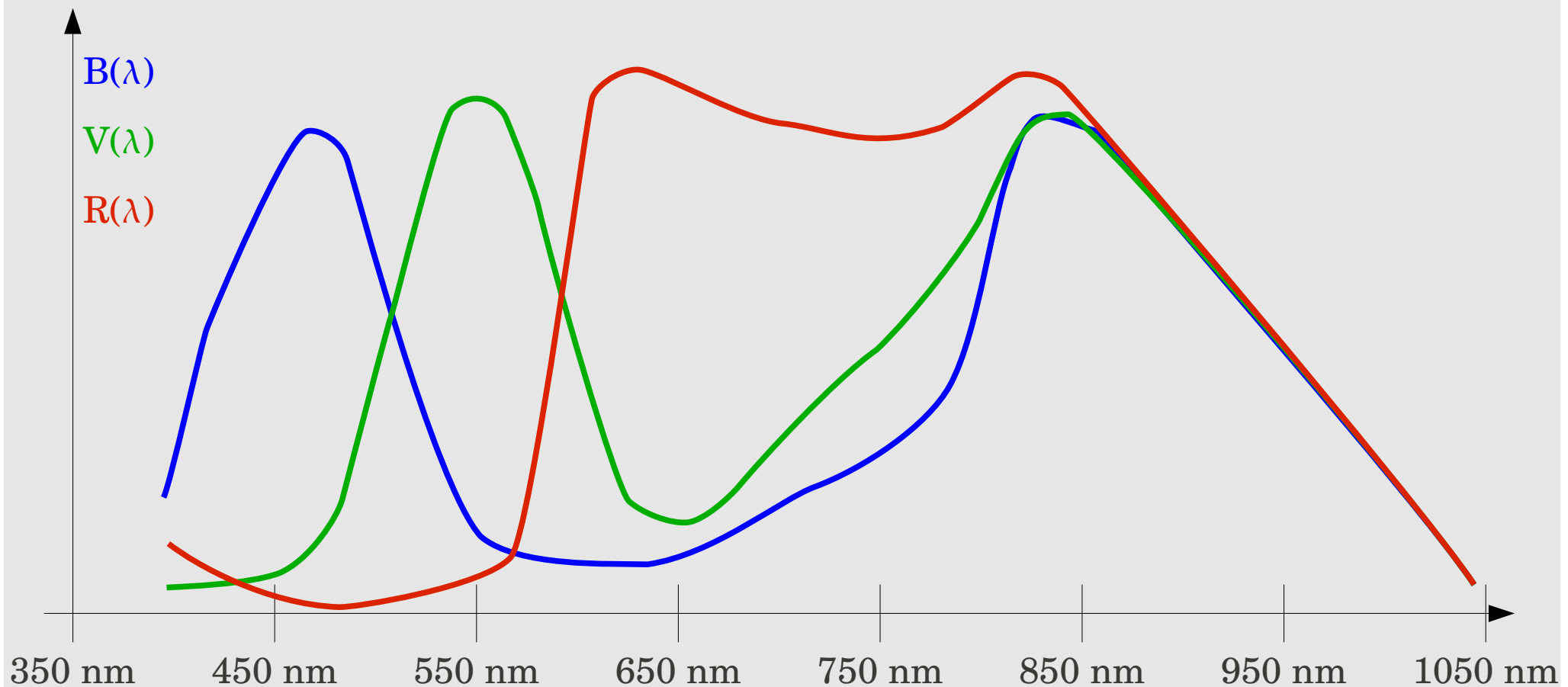
Capteur CMOS associé à une matrice de filtres couleur

CFA = Color Filter Array, arrangement de type **Bayer**

ligne	colonne											751			
	0	1	2	3	4	5	6	7								
0	R	V	R	V	R	V	R	V					R	V	R	V
1	V	B	V	B	V	B	V	B					V	B	V	B
2	R	V	R	V	R	V	R	V					R	V	R	V
3	V	B	V	B	V	B	V	B					V	B	V	B
....																
476	R	V	R	V	R	V	R	V					R	V	R	V
477	V	B	V	B	V	B	V	B					V	B	V	B
478	R	V	R	V	R	V	R	V					R	V	R	V
479	V	B	V	B	V	B	V	B					V	B	V	B

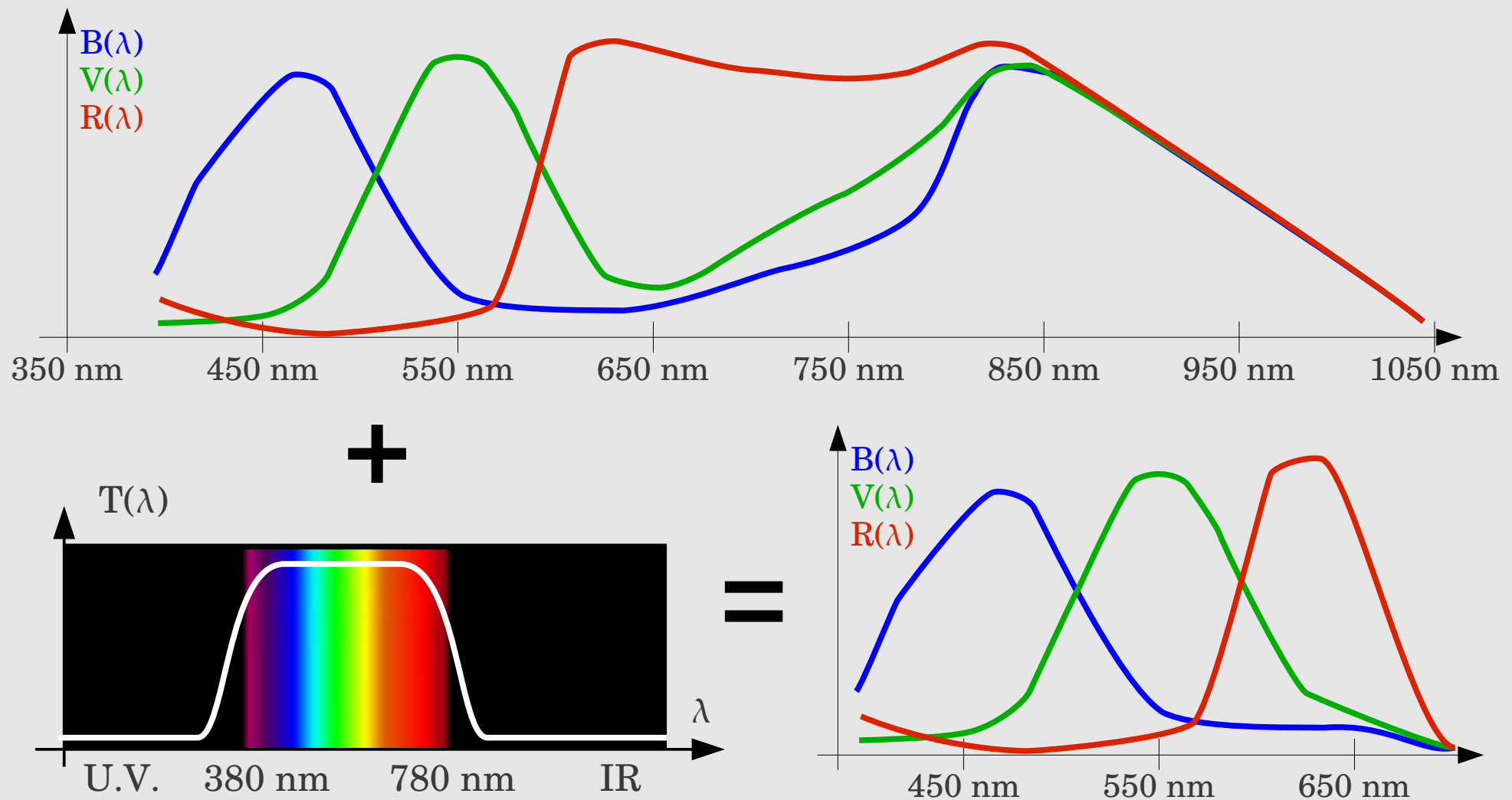
Capteur couleur ou infrarouge (1/3)

Courbes de réponse spectrale des 3 types de pixels



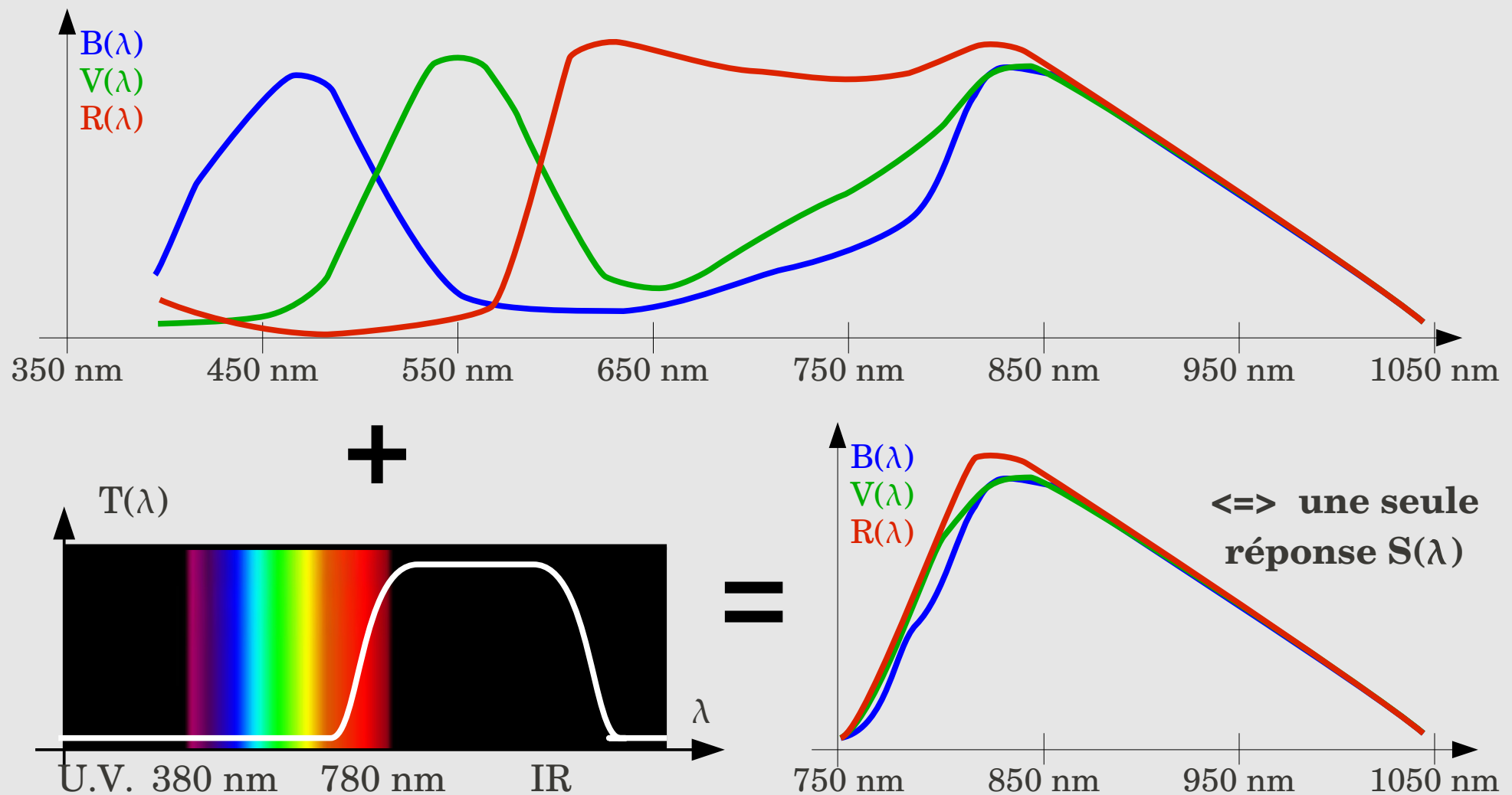
Capteur couleur ou infrarouge (2/3)

Association avec un filtre passe-bande visible



Capteur couleur ou infrarouge (3/3)

Association avec un filtre passe-bande infrarouge



Chaîne d'acquisition d'image

Fonction réalisée

transformation de l'**information lumineuse** représentative de la scène en une séquence d'**images numériques**.

Éléments constitutifs de la chaîne

objectif (KOWA LMVZ164) et **caméra** (IDS uEye 1225-C)

interface USB 2.0 et **pilote logiciel** associé (driver)

openCV (bibliothèque de traitement des images, C++)



Transfert du flux vidéo vers l'USB

Intégration de l'information lumineuse (capteur)

l'image projetée par l'objectif sur le capteur CMOS est **intégrée** pendant une durée fixée à l'avance (**temps d'intégration**).

après la période d'intégration, une **mémoire** intégrée au capteur contient les **mesures de l'énergie** réalisées pour chaque pixel.

Balayage des cellules du capteur

l'électronique de la caméra **balaye** la matrice de capteurs, ligne par ligne, et pixel par pixel dans chaque ligne.

l'**horloge pixel** (en MHz) fixe la cadence de balayage des pixels.

Flux vidéo

le **flux vidéo** contient les mesures, sous forme numérique.

à chaque top de l'horloge, une nouvelle **donnée** est transmise entre la caméra et l'ordinateur via le bus **USB**.

Acquisition du côté ordinateur depuis l'USB

« Périphérique » caméra

du point de vue de l'ordinateur, la vidéo est un flux de données numériques fournies par un **périphérique USB**.

pour gérer les particularités de de périphérique, il faut qu'un **pilote logiciel** adapté soit installé sur l'ordinateur.

Rôle du pilote logiciel

vérifie que la caméra et la connexion associée fonctionnent correctement via des fonctions de **diagnostic** et d'**initialisation**.

fixe les **paramètres** de fonctionnement de la caméra.

permet de transformer le flux vidéo en **tableaux de données** numériques représentant les images successives.

toutes ces fonctions sont disponibles dans une **bibliothèque**.

Fonctions du pilote logiciel uEye (SDK)

Généralités

fonctions appelées depuis un langage évolué (C++, Java, etc...)
caméra désignée par un **handle**, comme pour les fichiers en C++
convention pour le nom des fonctions : **is_NomDeFonction**
toutes les fonctions renvoient un **code d'erreur** (entier « int »).

Initialisation de la caméra

int is_InitCamera (HIDS* hCam, NULL);

HIDS est le type de données C++ correspondant au **handle**.

la fonction vérifie qu'une caméra IDS est connectée sur un port USB, initialise la caméra et retourne un handle (dans hCam).

handle = **premier paramètre** dans toutes les autres fonctions.

Fermeture du périphérique caméra

int is_ExitCamera (HIDS hCam);

Stockage des images en mémoire

Mode d'acquisition : *is_SetColorMode (HIDS hCam, int Mode);*

du côté caméra, indique comment les données doivent être **transférées** dans le flux vidéo.

du côté ordinateur, indique comment les données issues du flux vidéo doivent être **rangées** dans la mémoire.

	octet 0	octet 1	octet 2	octet 3	octet 4
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Format NdG	luminance n	luminance n+1	luminance n+2	luminance n+3	luminance n+4
Format RGB32	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	inutilisé	7 6 5 4 3 2 1 0
Format RGB16	2 1 0 4 3 2 1 0	4 3 2 1 0 5 4 3 2 1 0	4 3 2 1 0 5 4 3 2 1 0	4 3 2 1 0 5 4 3	2 1 0 4 3 2 1 0
Format BAYER	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
ligne suivante	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0

Autres fonctions utilisées durant le TP

is_SetPixelClock (HIDS hCam, int Clock);

définit la **fréquence** de l'horloge pixel (Clock exprimé en MHz)

is_SetFrameRate (HIDS hCam, double FPS, double newFPS);*

définit la **cadence** souhaitée d'acquisition des images, renvoie dans la variable *newFPS* la cadence effectivement fixée

is_SetExposureTime (HIDS hCam, double EXP, double newEXP);*

définit le **temps d'exposition** souhaité, renvoie dans la variable *newEXP* le temps d'exposition effectivement fixé

is_SetHardwareGain (HIDS hCam, int M, int R, int G, int B);

définit les **gains** (facteurs d'amplification des signaux). Les paramètres *M*, *R*, *G* et *B* prennent des valeurs comprises entre 0 et 100. Le **gain réel** dépend de la valeur fournie : 0 correspond à un gain de 1, la valeur 100 correspond à un gain de 5.

M fixe le **gain total**, *R*, *G* et *B* les gains par **composante couleur**

Gestion mémoire et affichage (1/2)

Principe

les images sont acquises par la bibliothèque du **pilote IDS**
elles sont affichées (et/ou traitées) par la bibliothèque **openCV**
les deux bibliothèques interagissent via une **mémoire partagée**

pilote IDS : *HIDS hCam*

la fonction *is_SetImageMem*
associe la mémoire *pcMemoryImage*
à la caméra, via le handle *hCam*

openCV : *IplImage* cvImage*

cvImage->imageData définit
l'adresse de la mémoire utilisée,
on lui affecte *pcMemoryImage*

mémoire partagée : *char* pcMemoryImage*

Gestion mémoire et affichage (2/2)

openCV (Open Source Computer Vision)

durant ce premier TP, la bibliothèque openCV est utilisée uniquement pour **afficher les images**

Fonctions utilisées

cvNamedWindow (char *name, CV_WINDOW_AUTOSIZE); crée

une fenêtre d'affichage pour les images, repérée par son nom

cvDestroyWindow (char * name); détruit la fenêtre d'affichage

cvShowImage (char *name, IplImage* cvImage); affiche la structure image dans la fenêtre d'affichage nommée *name*

int cvWaitKey (int delay); attend l'appui sur une touche pendant *delay* millisecondes, et renvoie le code de la touche

cvCreateImageHeader et ***cvReleaseImageHeader*** permettent d'initialiser puis de détruire une structure image openCV