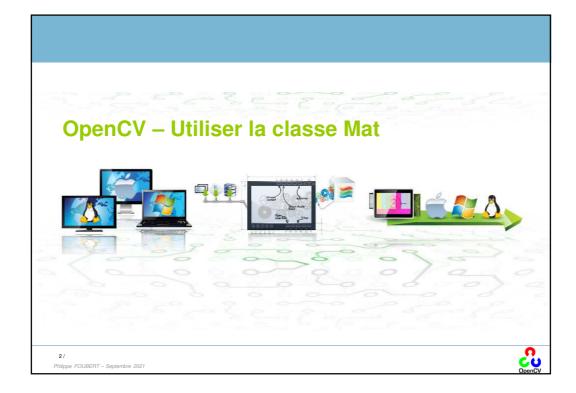


Philippe FOUBERT



# **Mat - Expressions**

Conventions: A et B sont des matrices, s est un Scalar, r est une valeur

Expression	Description
A+B, A-B, A+s, A-s, s-A, -A	Addition, soustraction, négation
A*r	Echelle
A.mul(B), A/B, r/A	Multiplication et division terme à terme
A.t()	Transposition
A cmpop B, A cmpop alpha, r cmpop A	Comparaison, avec cmpop : >, >=, ==, !=, <=, <
A logicop B, A logicop s, s logicop A, ~A	Opérations logiques, avec logicop : &,  , ^
min(A, B), min(A, r), max(A, B), max(A, r)	Minimum et maximum
abs(A)	Valeur absolue
A.cross(B), A.dot(B)	Produit vectoriel, produit scalaire

Philippe FOUBERT - Septembre 202



# Mat - Manipulation d'un pixel

- → Soit m une image composée d'un seul canal de type float : m.at<float>(idx\_row, idx\_col) = newval;
- → Soit n une image composée de 3 canaux de type uchar :

```
n.at<cv::Vec3b>(idx_row, idx_col)[0] = newval_1; // Bleu
n.at<cv::Vec3b>(idx_row, idx_col)[1] = newval_2; // Vert
n.at<cv::Vec3b>(idx_row, idx_col)[2] = newval_3; // Rouge
```

4 /

CO

## Mat - Parcourir une image

→ Soit m une image composée d'un seul canal de type float :

```
for (int idx_row = 0; idx_row < m.rows; idx_row++) {
   float* ptr = m.ptr<float>(idx_row);
   for (int idx_col = 0; idx_col < m.cols; idx_col ++)
        ptr[idx_col] = newval;
}</pre>
```



→ Soit n une image composée de 3 canaux de type uchar :

```
for (int idx_row = 0; idx_row < n.rows; idx_row++) {
   for (int idx_col = 0; idx_col < n.cols; idx_col++) {
        n.at<cv::Vec3b>(idx_row, idx_col)[0] = newval_1; // Bleu
        n.at<cv::Vec3b>(idx_row, idx_col)[1] = newval_2; // Vert
        n.at<cv::Vec3b>(idx_row, idx_col)[2] = newval_3; // Rouge
   }
}
```

Philippa EOUBERT - Sentembre 2023



### Mat – Parcourir une image (plus efficace)

→ Soit m une image composée d'un seul canal de type float :

```
cv::MatIterator_<float> it, end;
for(it = m.begin<float>(), end = m.end<float>();
   it != end; ++it) {
   *it = newval;
}
```

→ Soit n une image composée de 3 canaux de type uchar :

```
cv::MatIterator_<cv::Vec3b> it, end;
for(it = n.begin<cv::Vec3b>(), end = n.end<cv::Vec3b>();
   it != end; ++it) {
   (*it)[0] = newval_1;
   (*it)[1] = newval_2;
   (*it)[2] = newval_3;
}
```

6/

င္မွပ္မ

# Mat – Précautions à prendre

OpenCV limite le plus possible les opérations mémoire :

```
cv::Mat A = cv:: Mat::zeros(2, 3, CV_64F);
 cv::Mat B = cv:: Mat::ones(2, 3, CV_64F);
A = B
```

[...]

B = cv:: Scalar(2);std::cout << A;

Que s'est-il passé?

A pointe vers le contenu de B

Que souhaitait-on? Copier le contenu de B dans A

Comment procéder ? A = B.clone();

