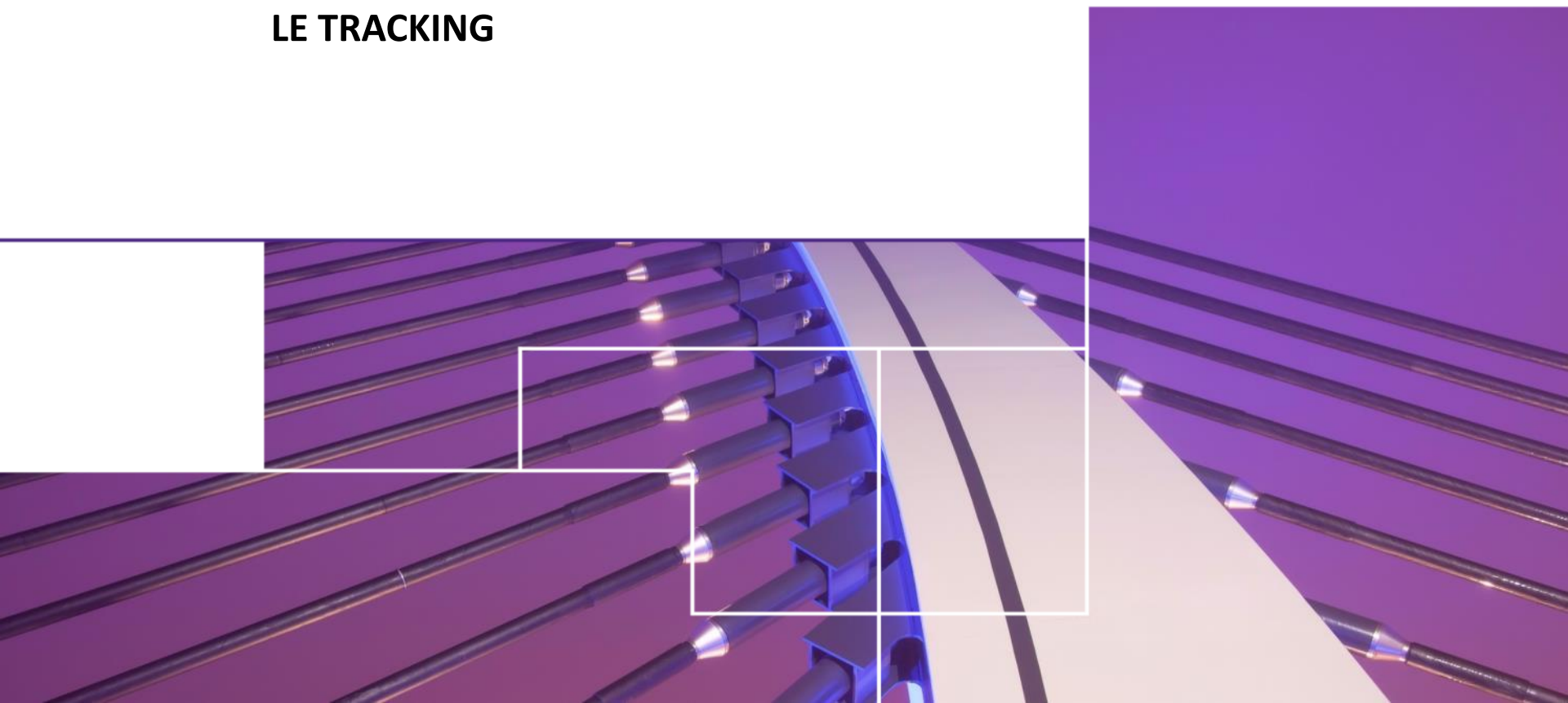




LE TRACKING





01

LE TRACKING

QU'EST CE QUE LE TRACKING ?

DÉFINITION DU TRACKING

Object tracking : suivre le déplacement d'un **même objet** dans une vidéo, séquence d'images, ...

En computer vision : dépendant d'une **détection préalable** !

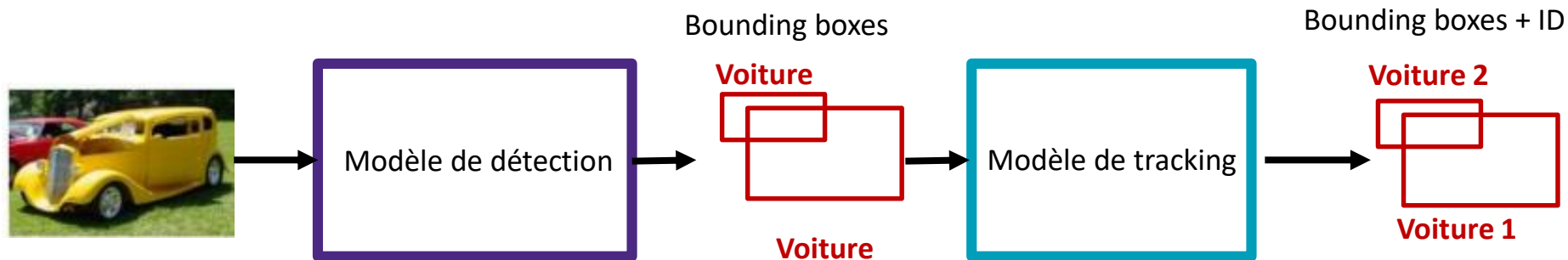
- Dépendant d'une détection
 - Bonne détection : tracking facilité
 - Mauvaise détection : tracking impaired
- Peut être utilisée dans une tâche de détection pour améliorer celle-ci
 - Pour corriger les erreurs de la détection et prendre en compte l'aspect séquentiel

QU'EST CE QUE LE TRACKING ?

DÉFINITION DU TRACKING

Object tracking : suivre le déplacement d'un **même objet** dans une vidéo, séquence d'images, ...

En computer vision : dépendant d'une **détection préalable** !

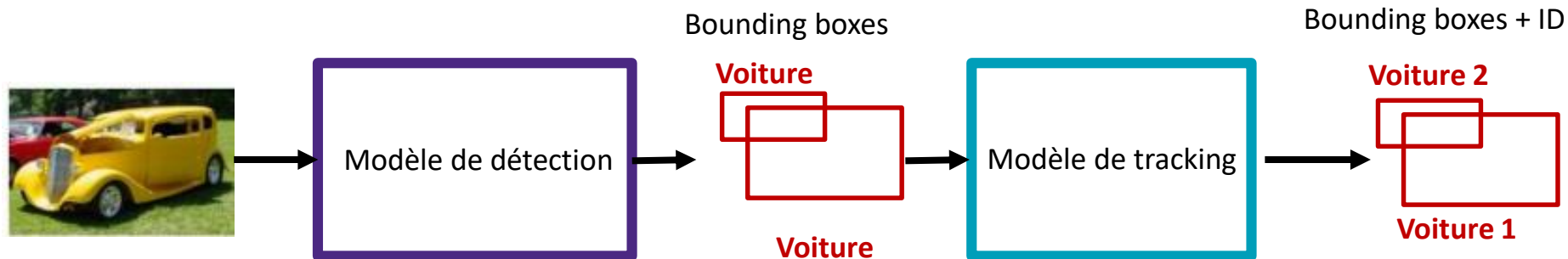


QU'EST CE QUE LE TRACKING ?

DÉFINITION DU TRACKING

Object tracking : suivre le déplacement d'un **même objet** dans une vidéo, séquence d'images, ...

En computer vision : dépendant d'une **détection préalable** !

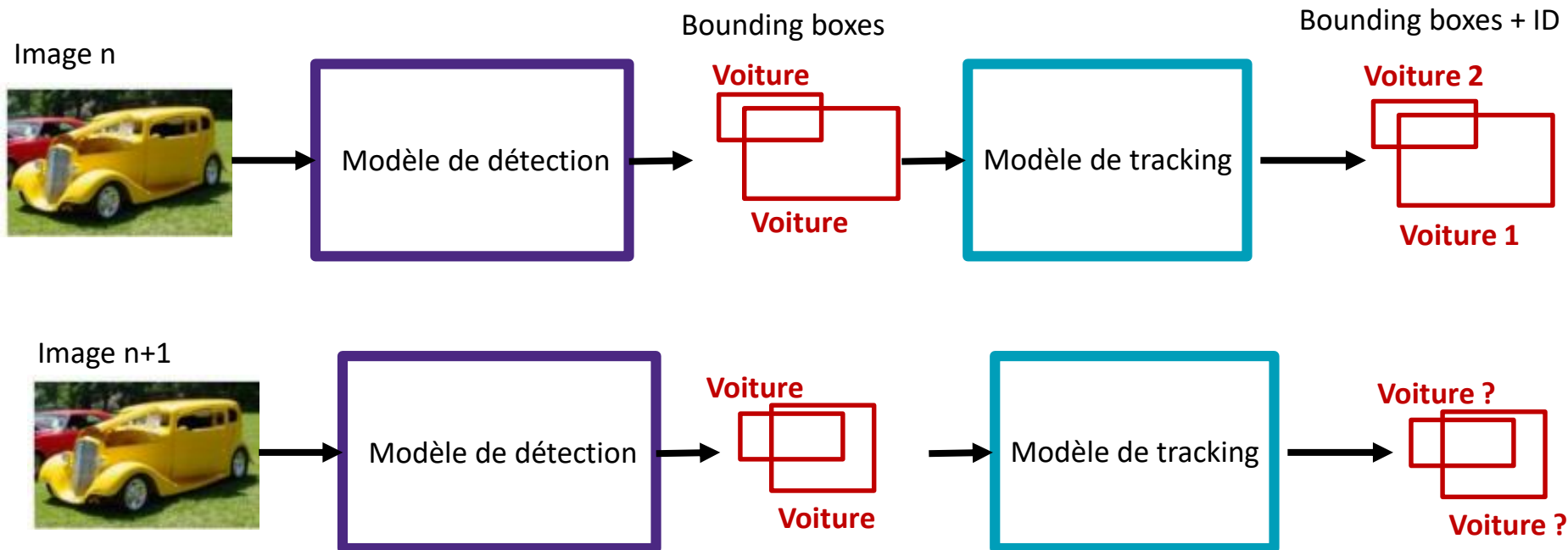


- Attribue à chaque objet un **identifiant** pour le suivre
 - Impact au niveau de la labélisation
 - Il nous faut en plus des labels de la détection (classe, bounding box, ...) l'id de l'objet pour chaque détection.

QU'EST CE QUE LE TRACKING ?

DÉFINITION DU TRACKING

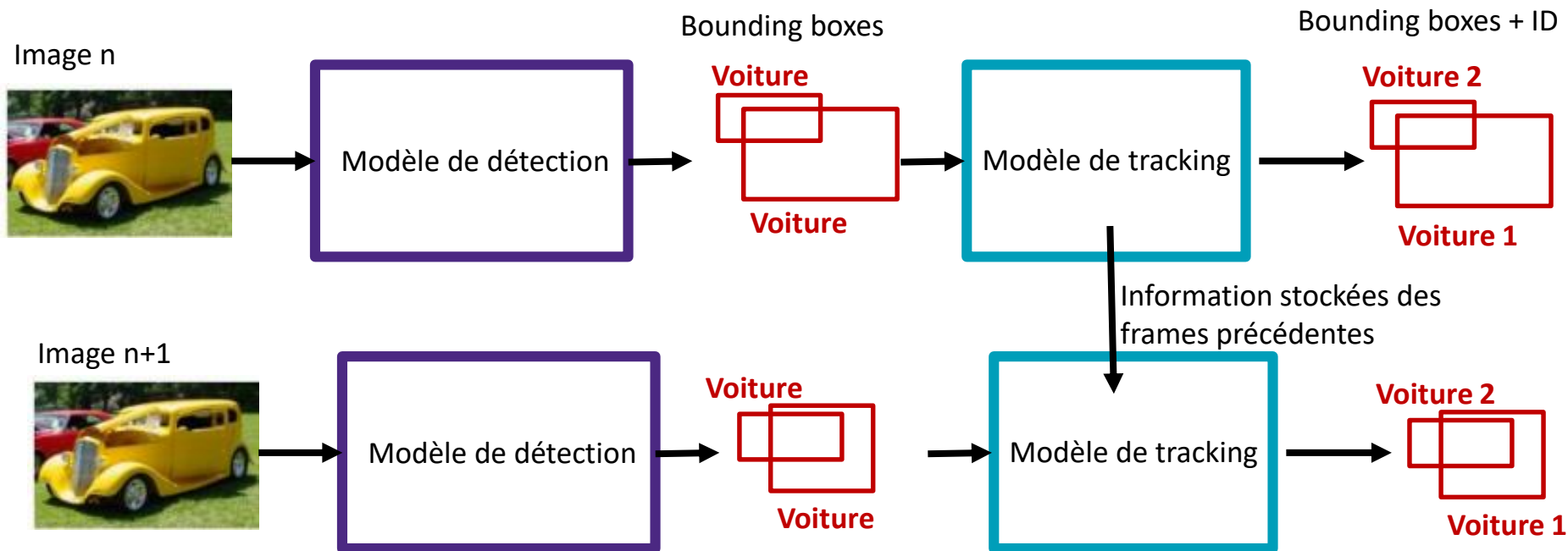
Object tracking : suivre le déplacement d'un **même objet** dans une vidéo, séquence d'images, ...



QU'EST CE QUE LE TRACKING ?

DÉFINITION DU TRACKING

Object tracking : suivre le déplacement d'un **même objet** dans une vidéo, séquence d'images, ...

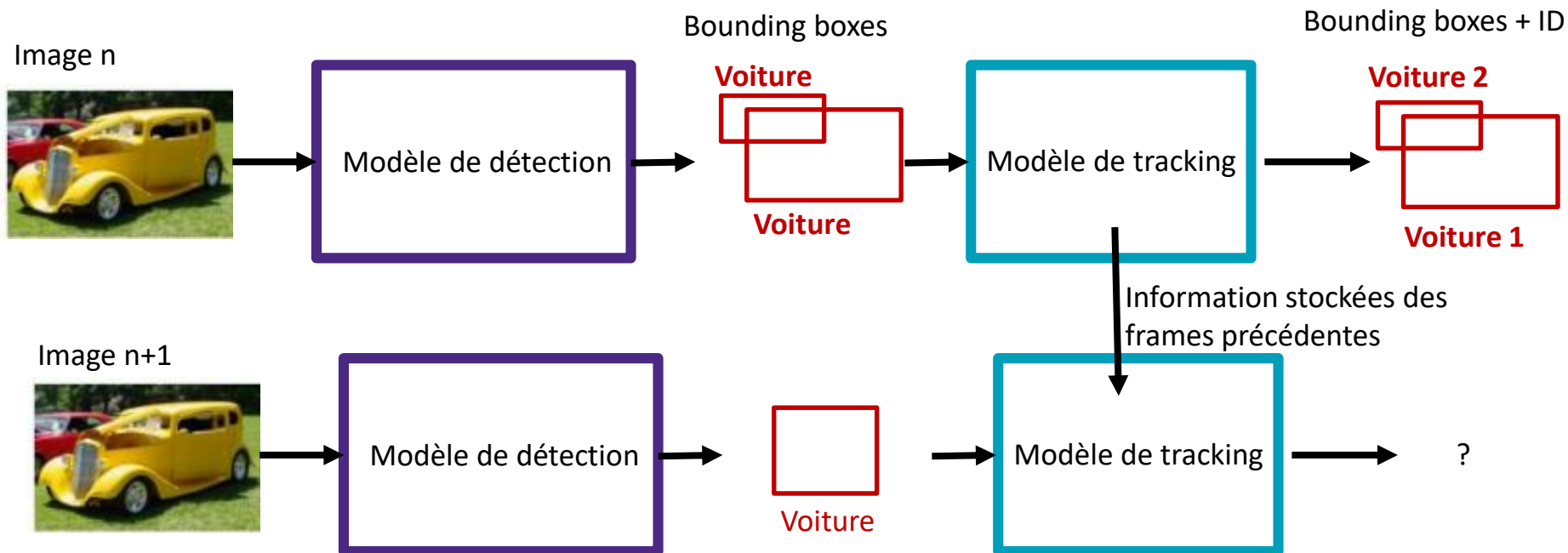


PROPRIÉTÉS

Dans ce genre d'architecture, le tracking est **dépendant de la qualité de la détection**

- Bonne détection : tracking facilité
- Mauvaise détection : tracking impaired

Le tracking peut être utilisé dans les tâches de détection pour **améliorer les performances** de celle-ci.



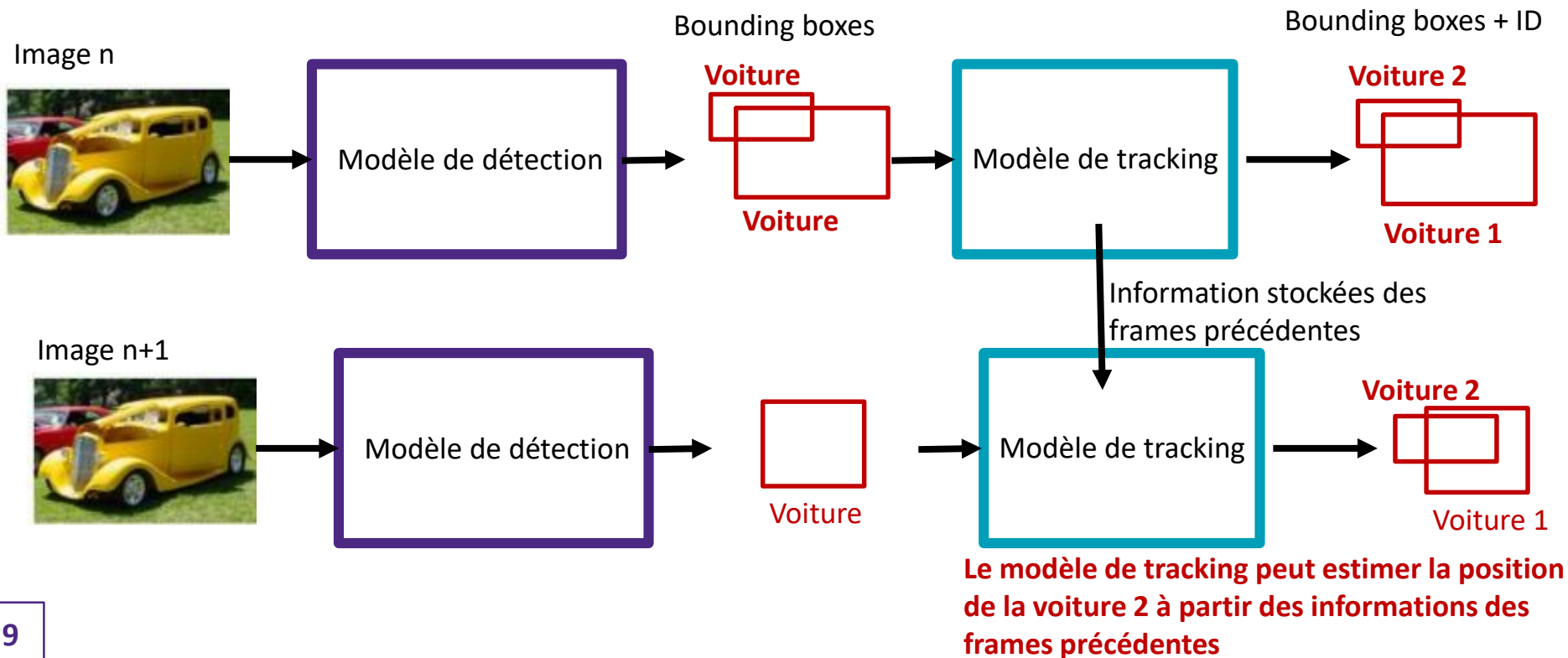
Le modèle de détection n'a pas détecté la voiture rouge sur cette frame !

PROPRIÉTÉS

Dans ce genre d'architecture, le tracking est **dépendant de la qualité de la détection**

- Bonne détection : tracking facilité
- Mauvaise détection : tracking impaired

Le tracking peut être utilisé dans les tâches de détection pour **améliorer les performances** de celle-ci.



Algorithme de tracking « naïf » : **l'overlap**.

- Si deux boîtes aux images n et $n+1$ ont une intersection (IoU) non vides, elles correspondent au même objet

Avantage :

- Rapide et simple à utiliser
- S'il y a peu d'objets dans la scène et se déplaçant lentement par rapport au framerate.

Inconvénient :

- Ne marche que si la détection bien fonctionnée
 - Si le modèle de détection à manqué l'objet sur une frame ?
- Ne gère pas les cas complexes :
 - Si deux cibles potentielles à $n+1$ overlapent la même boîte à n ?
 - Si deux boîtes à n overlapent la même boîte à $n+1$?
 - Si deux personnes marche cote à cote et inversent de position ?
 - Si une personne disparaît dans une ou plusieurs frame ?
 - Obfuscation, passage derrière un obstacle, ...

FILTRE DE KALMAN

Le filtre de kalman : un algorithme très utilisé en robotique.

Idée :

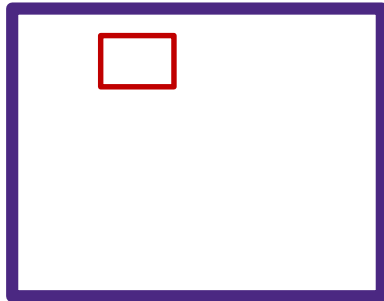
- A chaque pas de temps, mes capteurs me remontent des informations
 - Celles-ci peuvent contenir des erreurs.
- Je possède des informations des instants précédents
 - Celles-ci également peuvent être erreurs.

Qui est-ce que je dois croire si les mesures des capteurs diffèrent de ce à quoi je m'attends au vu des mesures précédentes ?

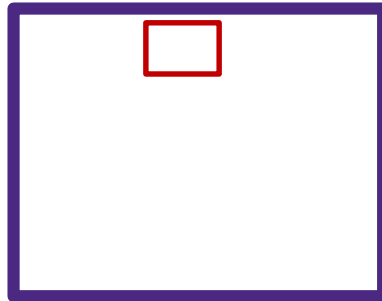
Frame n-3



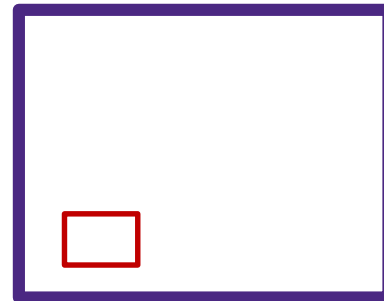
Frame n-2



Frame n-1



Frame n



Mon objet est-il
réellement ici ?

FILTRE DE KALMAN

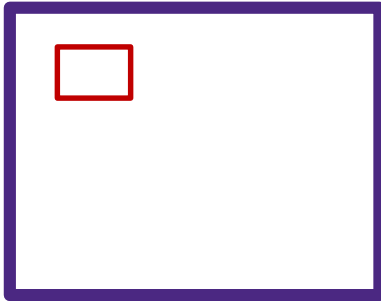
Le filtre de kalman : un algorithme très utilisé en robotique.

Idée :

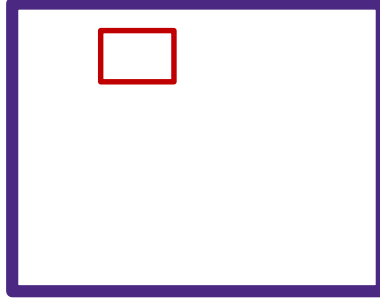
- A chaque pas de temps, mes capteurs me remontent des informations
 - Celles-ci peuvent contenir des erreurs.
- Je possède des informations des instants précédents
 - Celles-ci également peuvent être erreurs.

Qui est-ce que je dois croire si les mesures des capteurs diffèrent de ce à quoi je m'attends au vu des mesures précédentes ?

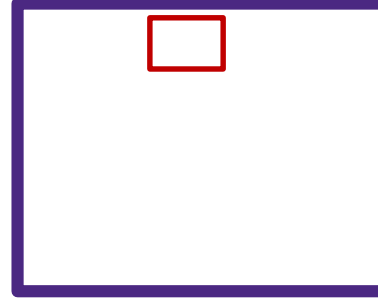
Frame n-3



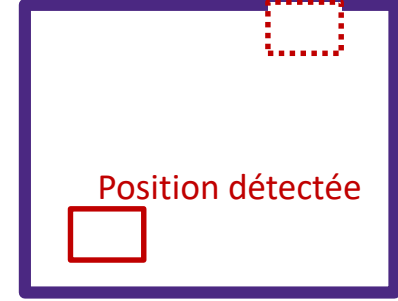
Frame n-2



Frame n-1



Frame n



Position réelle

Position détectée

Hypothèse 1 : La détection du capteur à la frame n est erronée

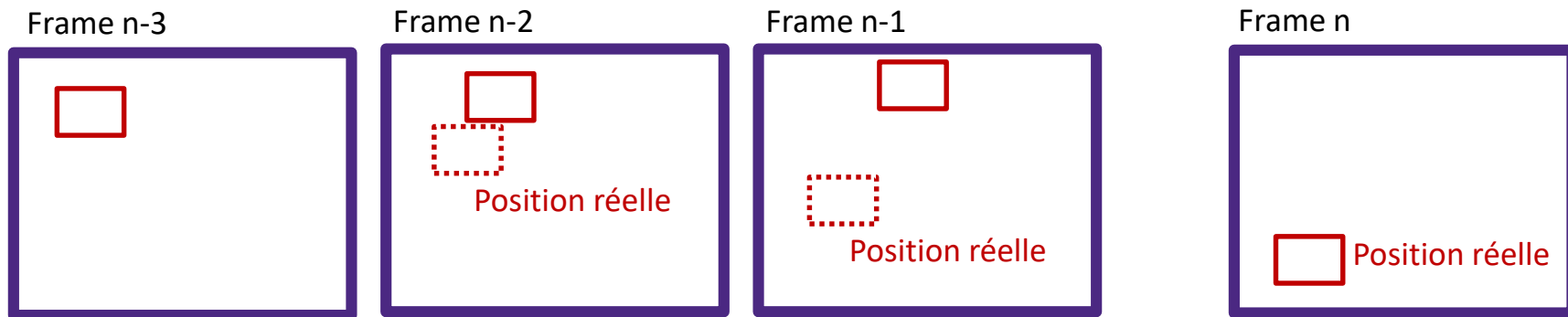
FILTRE DE KALMAN

Le filtre de kalman : un algorithme très utilisé en robotique.

Idée :

- A chaque pas de temps, mes capteurs me remontent des informations
 - Celles-ci peuvent contenir des erreurs.
- Je possède des informations des instants précédents
 - Celles-ci également peuvent être erreurs.

Qui est-ce que je dois croire si les mesures des capteurs diffèrent de ce à quoi je m'attends au vu des mesures précédentes ?



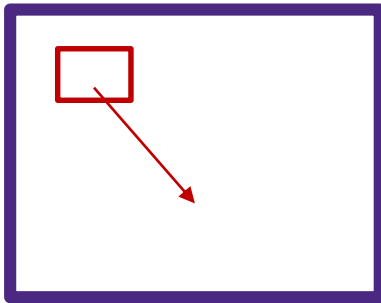
Hypothèse 2 : La détection du capteur à la frame est bonne, c'est mes états précédents qui étaient faux

FILTRE DE KALMAN

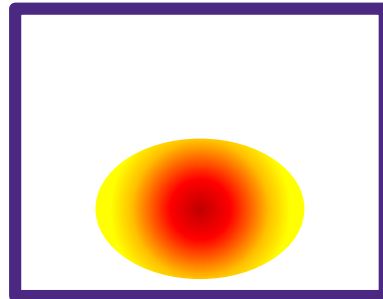
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité

Frame n-1



Frame n



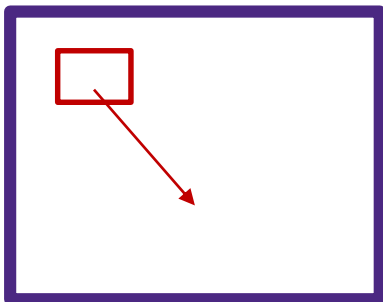
Position estimée au vu des mesures précédentes

FILTRE DE KALMAN

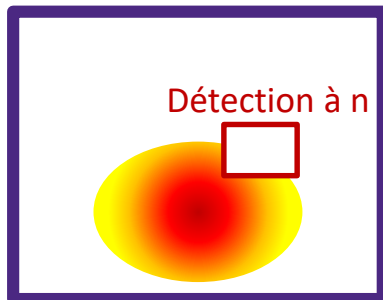
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation

Frame n-1



Frame n



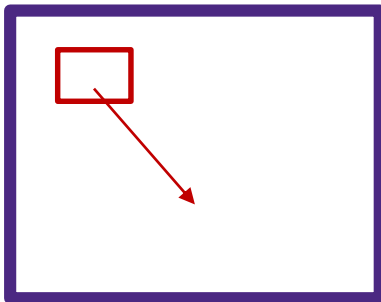
Position estimée au vu des mesures précédentes

FILTRE DE KALMAN

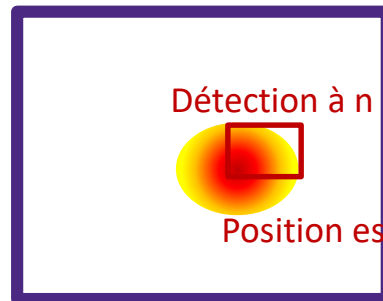
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation

Frame n-1



Frame n



Détection à n

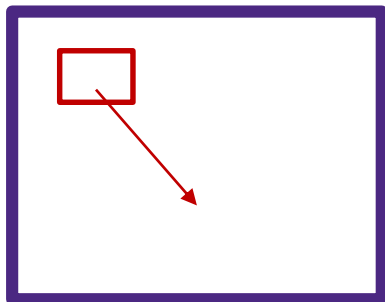
Position estimée au vu des mesures précédentes

FILTRE DE KALMAN

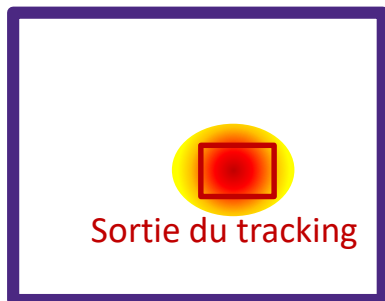
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation
 - Suivant la fiabilité estimée du capteur (pour nous : confiance du modèle dans sa prédiction), les mesures de celui-ci vont avoir plus ou moins d'importance dans la décision finale

Frame n-1



Frame n



FILTRE DE KALMAN

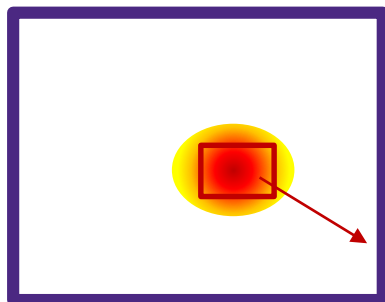
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation
 - Suivant la fiabilité estimée du capteur (pour nous : confiance du modèle dans sa prédiction), les mesures de celui-ci vont avoir plus ou moins d'importance dans la décision finale

Frame n-1



Frame n



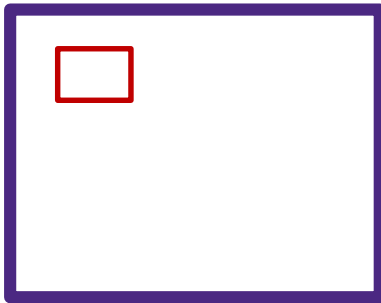
Nouvelles vitesses
et accélération estimées

FILTRE DE KALMAN

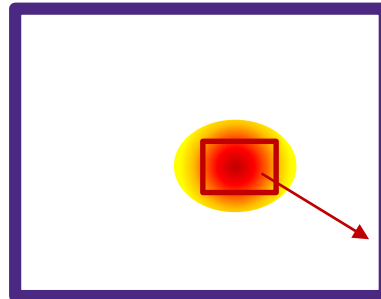
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation
 - Suivant la fiabilité estimée du capteur (pour nous : confiance du modèle dans sa prédiction), les mesures de celui-ci vont avoir plus ou moins d'importance dans la décision finale

Frame n-1



Frame n



Frame n+1



Position estimée

FILTRE DE KALMAN

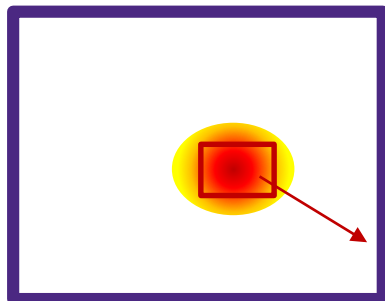
Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation
 - Suivant la fiabilité estimée du capteur (pour nous : confiance du modèle dans sa prédiction), les mesures de celui-ci vont avoir plus ou moins d'importance dans la décision finale

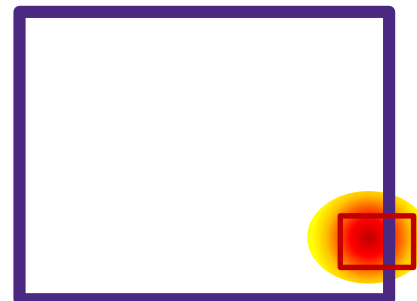
Frame n-1



Frame n



Frame n+1

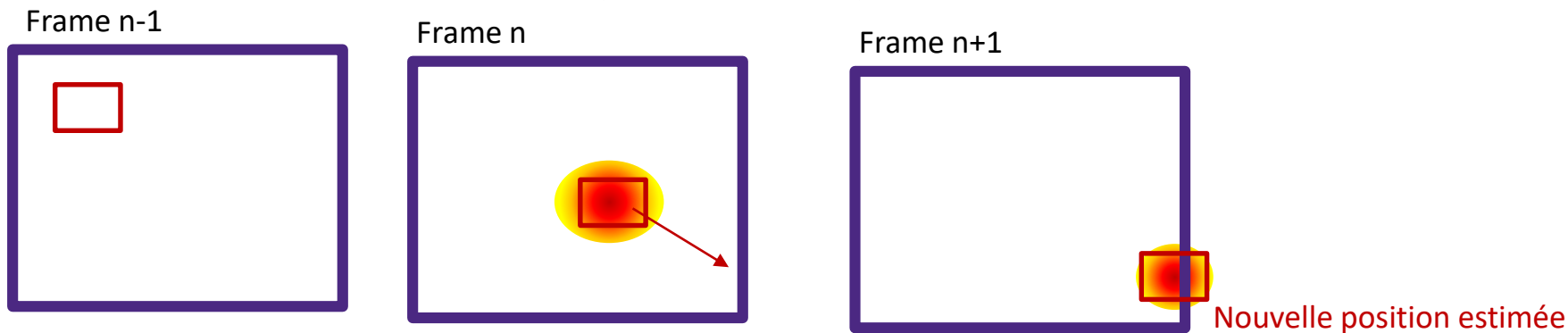


Nouvelle mesure

FILTRE DE KALMAN

Le filtre de KALMAN

- Estime la valeurs d'un certain nombre de variable en fonction des mesures précédentes (position, vitesse, accélération, ...)
 - Sous forme de densité de probabilité
- Puis prend en compte les nouvelles mesures (détection) pour raffiner son estimation
 - Suivant la fiabilité estimée du capteur (pour nous : confiance du modèle dans sa prédiction), les mesures de celui-ci vont avoir plus ou moins d'importance dans la décision finale

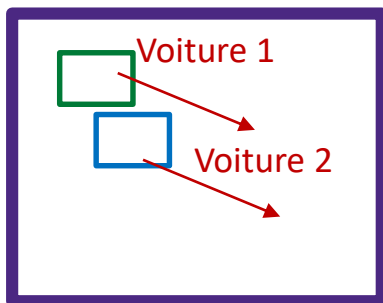


FILTRE DE KALMAN

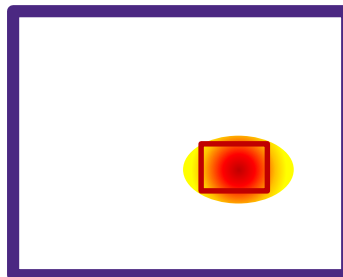
Le filtre de KALMAN

- Permet de gérer une absence de détection
- Prend en compte la vitesse, l'accélération, mais aussi souvent la taille de la boîte ou son ratio hauteur/largeur pour estimer quelle est la boîte la plus crédible
- Gère uniquement les boîtes une à une, et **ne prend pas en compte les autres détections** pour affiner son estimation
 - MoT : Multi-object tracking

Frame n



Frame n+1



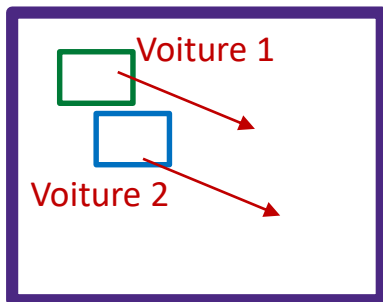
La probabilité dépend aussi des autres détections !

FILTRE DE KALMAN

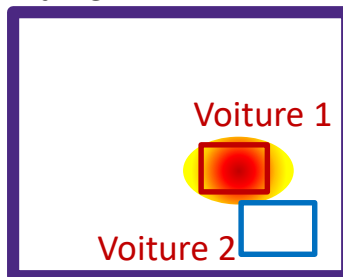
Le filtre de KALMAN

- Permet de gérer une absence de détection
- Prend en compte la vitesse, l'accélération, mais aussi souvent la taille de la boîte ou son ratio hauteur/largeur pour estimer quelle est la boîte la plus crédible
- Gère uniquement les boîtes une à une, et **ne prend pas en compte les autres détections** pour affiner son estimation
 - MoT : Multi-object tracking

Frame n



Frame n+1



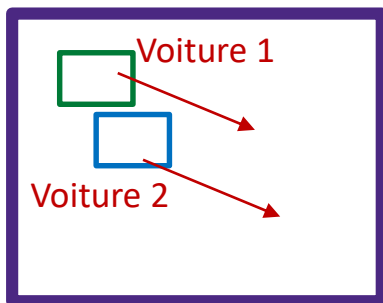
Si la voiture 2 est détectée
ici, c'est probable

FILTRE DE KALMAN

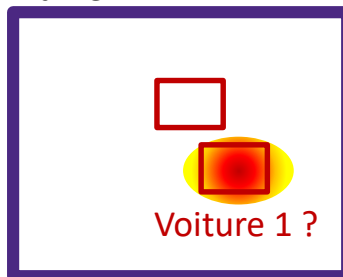
Le filtre de KALMAN

- Permet de gérer une absence de détection
- Prend en compte la vitesse, l'accélération, mais aussi souvent la taille de la boîte ou son ratio hauteur/largeur pour estimer quelle est la boîte la plus crédible
- Gère uniquement les boîtes une à une, et **ne prend pas en compte les autres détections** pour affiner son estimation
 - MoT : Multi-object tracking

Frame n



Frame n+1



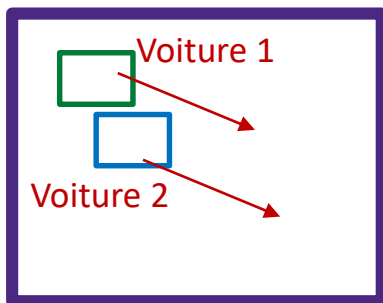
Si au contraire la seconde détection est ici, il est **peu probable** que la détection la plus proche corresponde à la voiture 1

FILTRE DE KALMAN

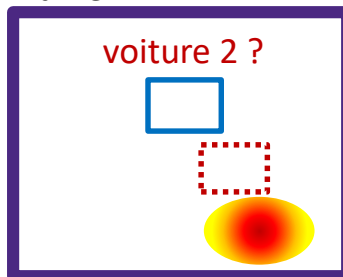
Le filtre de KALMAN

- Permet de gérer une absence de détection
- Prend en compte la vitesse, l'accélération, mais aussi souvent la taille de la boîte ou son ratio hauteur/largeur pour estimer quelle est la boîte la plus crédible
- Gère uniquement les boîtes une à une, et **ne prend pas en compte les autres détections** pour affiner son estimation
 - MoT : Multi-object tracking

Frame n



Frame n+1



Car sinon l'erreur sur la voiture 2 serait très importante

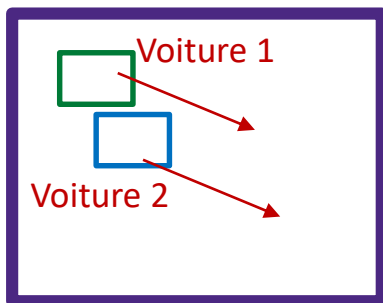
Voiture 2 estimée

FILTRE DE KALMAN

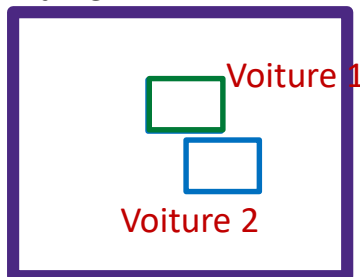
Le filtre de KALMAN

- Permet de gérer une absence de détection
- Prend en compte la vitesse, l'accélération, mais aussi souvent la taille de la boîte ou son ratio hauteur/largeur pour estimer quelle est la boîte la plus crédible
- Gère uniquement les boîtes une à une, et **ne prend pas en compte les autres détections** pour affiner son estimation
 - MoT : Multi-object tracking

Frame n



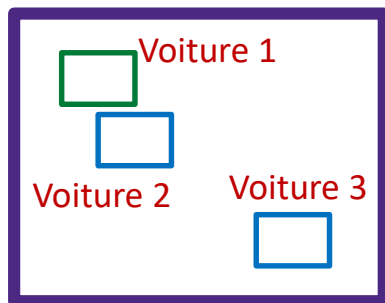
Frame n+1



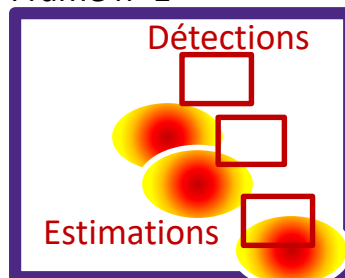
La probabilité du tracking d'un objet dépend également du tracking des autres objets !

Methode SORT : utilise KALMAN augmenté d'un algorithme Hungarian pour le MOT

Frame n



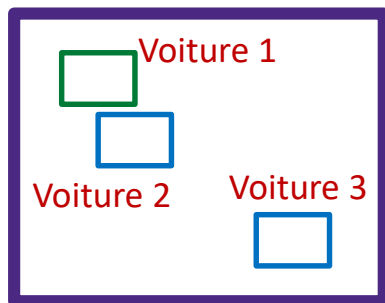
Frame n+1



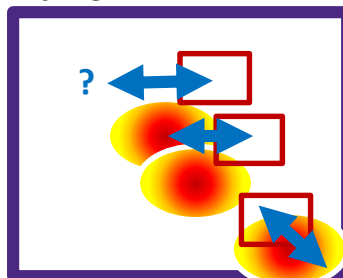
Problème d'attribution : quelle détection attribuer à chaque estimation ?

Methode SORT : utilise KALMAN augmenté d'un algorithme Hungarian pour le MOT

Frame n



Frame n+1

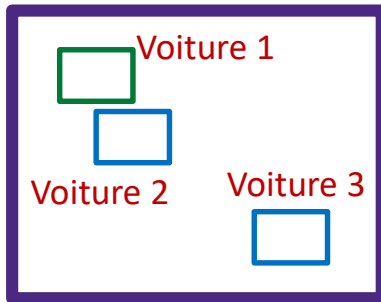


Ce n'est pas la plus proche

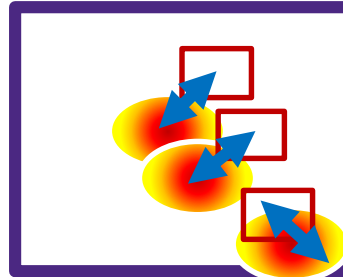
Methode SORT : utilise KALMAN augmenté d'un algorithme Hungarian pour le MOT

- Idée de SORT : attribuer à chaque estimation la détection qui minimise **la somme des différences** entre chaque estimations et sa détection attribuée
 - Problème NP complet : bien trop long à calculer
 - **L'algorithme Hongrois** est un heuristique performant permettant de résoudre ce problème d'attribution.

Frame n



Frame n+1

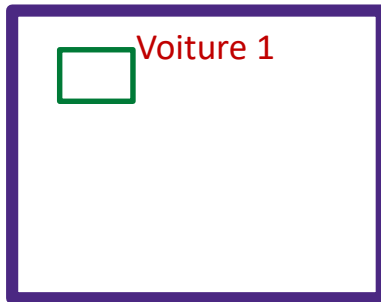


L'association estimations-détection est effectuée par l'algorithme hongrois (hungarian algorithm)

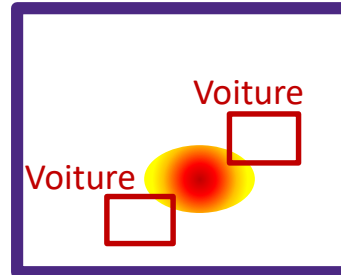
Les limites :

- SORT ne prend en compte que la position, vitesse, accélération, ratio, ... des bounding boxes pour effectuer le tracking

Frame n



Frame n+1



Comment départager quelle est la bonne détection ?

- Cette situation arrive souvent :
 - Occlusion, obstacle, nouveaux objets...

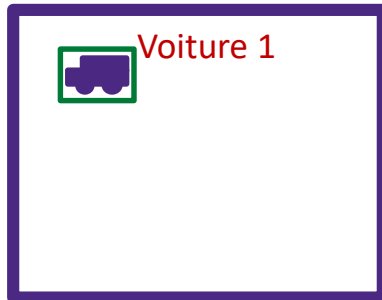
Les limites :

- SORT ne prend en compte que la position, vitesse, acceleration, ratio, ... des bounding boxes pour effectuer le tracking
 - Comment départager deux candidats aussi crédible l'un que l'autre ?
 - Intérêt du contenu de la boîte

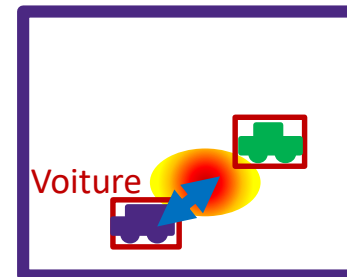
DeepSort rajoute une partie deep learning qui extrait des **features** à l'intérieur des bounding box.

- Des features similaires à celles que du transfert learning, i.e. extrait par une backone entraînée sur une autre tâche.
- L'association effectuée par l'algorithme hongrois se fait à partir de la **distance** des bounding boxes prédite et mesurée, mais aussi sur **la différence d'apparence de leur contenu**
 - L'algorithme hongrois est effectué sur la somme de ces deux distances

Frame n



Frame n+1

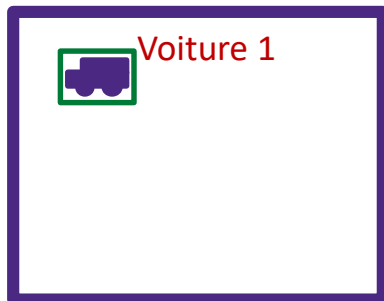


Deepsort sait départager, car le **contenu** de l'une des box est plus proche du contenu de la bounding box dans la frame n

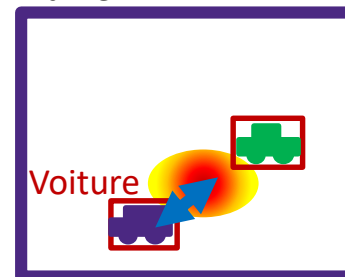
Deepsort :

- **Plus couteux:**
 - Une extraction de feature de Deep Learning par prediction dans la frame n+1
 - Scale linéairement en fonction du nombre de predictions de la detections.
 - Mais sur des images de faibles résolution, ce qui reste rapide
- **Indépendant de l'algorithme de détection**
 - On peut prendre un algorithme de détection rapide ou efficace suivant le besoin
 - Détection en une ou deux passes, réseau tiny ou non, ...
- **Meilleure performance**
- **Reste limité par les performances de l'algorithme de détection**

Frame n



Frame n+1



Deepsort sait départager, car le **contenu** de l'une des box est plus proche du contenu de la bounding box dans la frame n