

# Calcul auto-organisé : cas du tracking multi-caméra

Adrien Fois

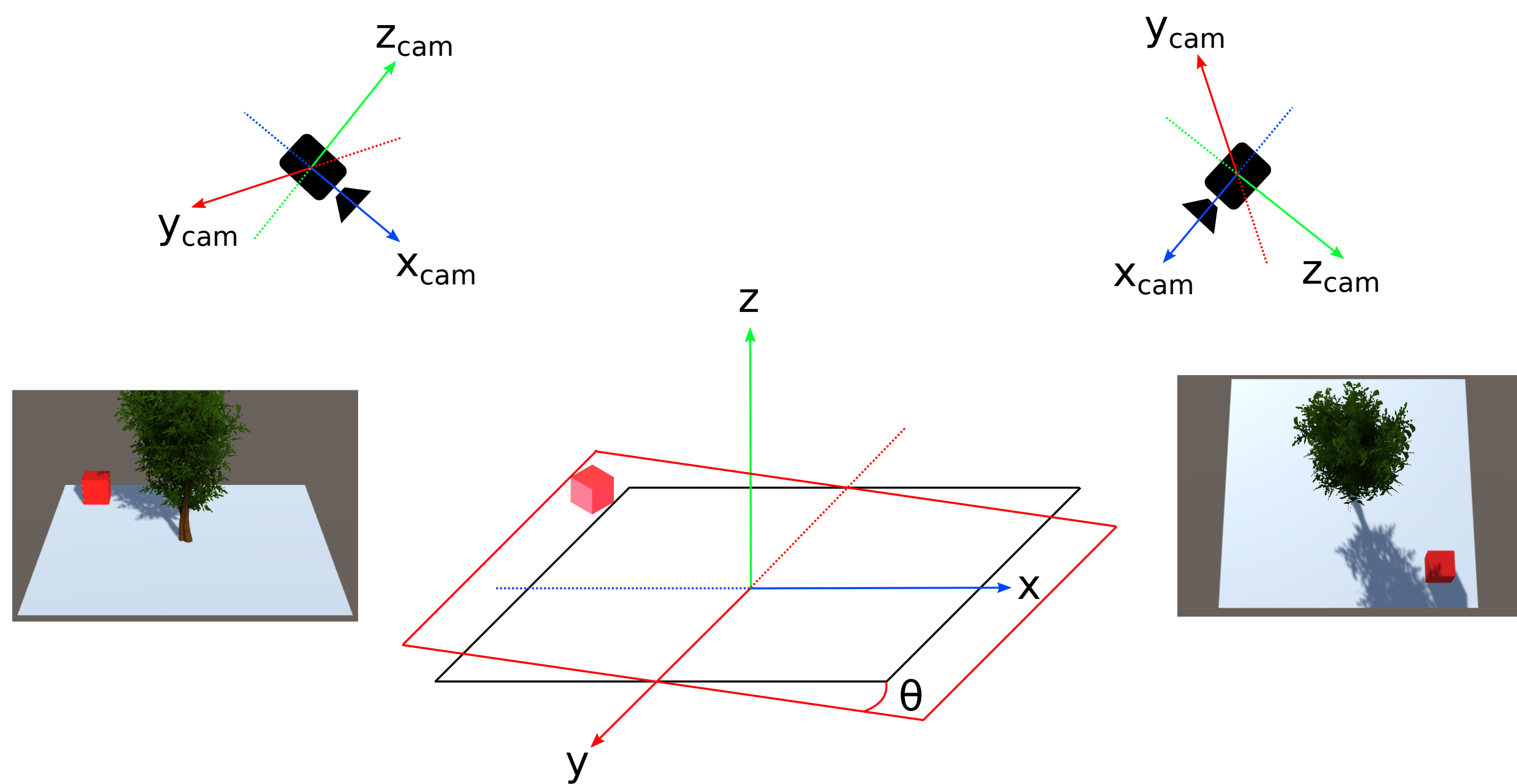
Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)

adrien.fois@loria.fr

## Abstract

Le but de ce travail est de tester les performances d'une allocation de ressources auto-organisée et bio-inspirée [1]. L'application considérée pour son caractère démonstratif est l'estimation de la position d'un objet en mouvement par deux réseaux de neurones, chacun traitant le flux vidéo d'une caméra. Usuellement, le nombre de neurones affecté à chaque réseau neuronal est une quantité constante. On teste ici, pour la première fois, l'effet d'une allocation dynamique de neurones sur la précision de l'estimation de la position d'un objet suivi. En outre, cette allocation de ressources se fait de façon auto-organisée en fonction de la complexité des flux entrants. L'hypothèse de travail est qu'une allocation de ressources auto-organisée, s'adaptant en continu aux fluctuations environnementales, peut améliorer l'erreur de suivi.

## 1. Montage expérimental



L'objet à suivre est un cube rouge en mouvement sur un plan, sa position dans son référentiel local est donnée par  $x, y$ . Un changement, sur une échelle temporelle plus lente, de la dynamique de l'environnement est modélisé par la variation de l'angle de rotation du plan  $\theta$ .

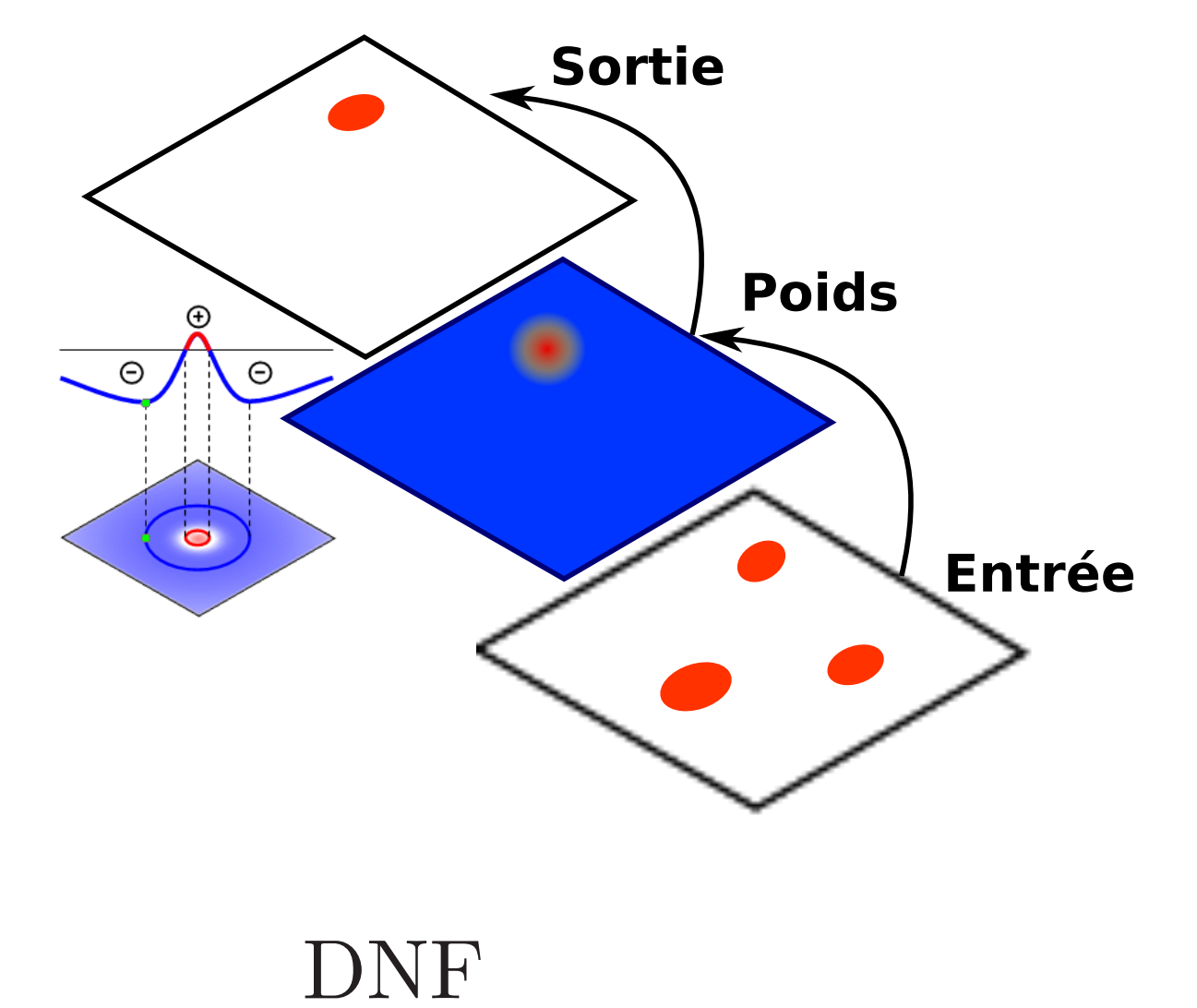
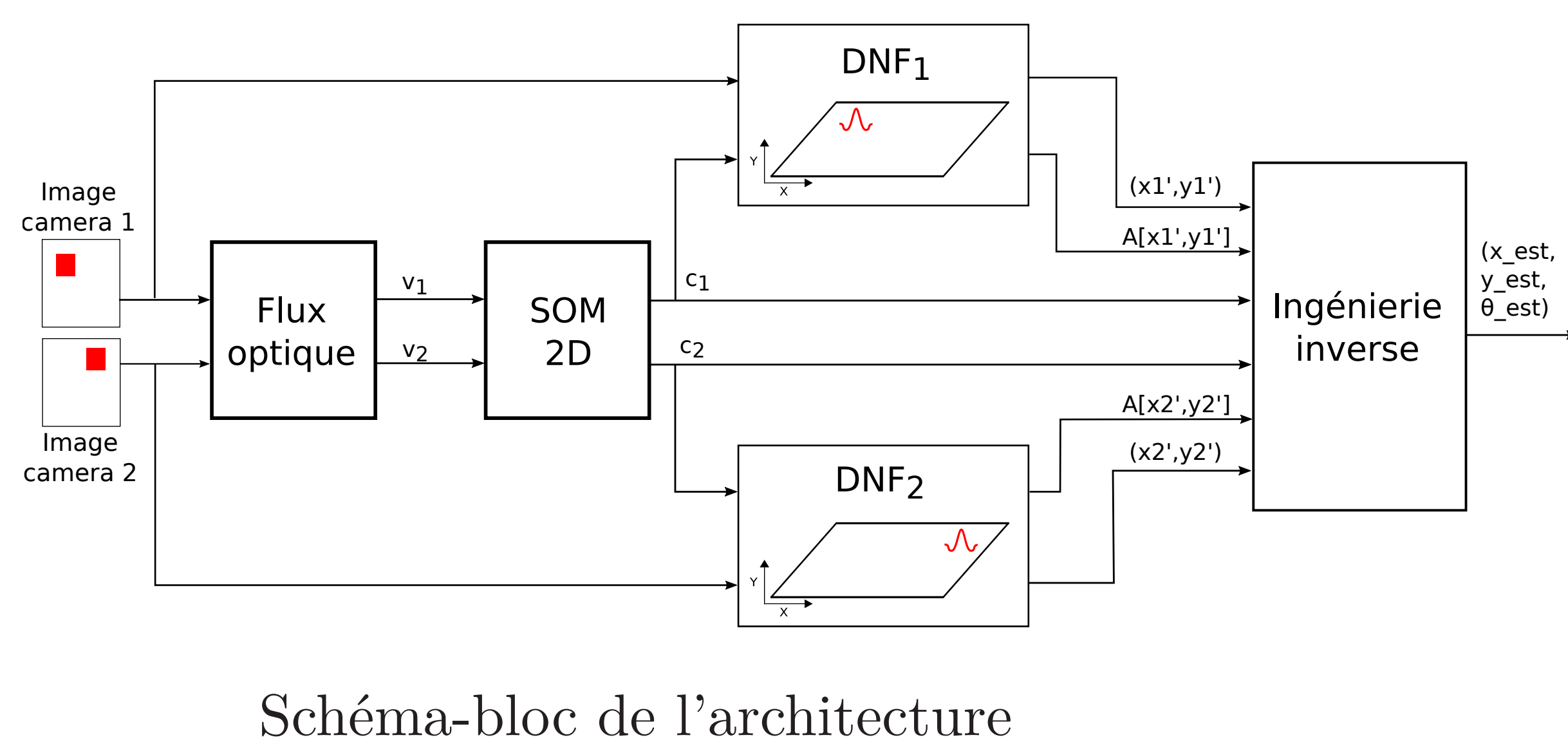
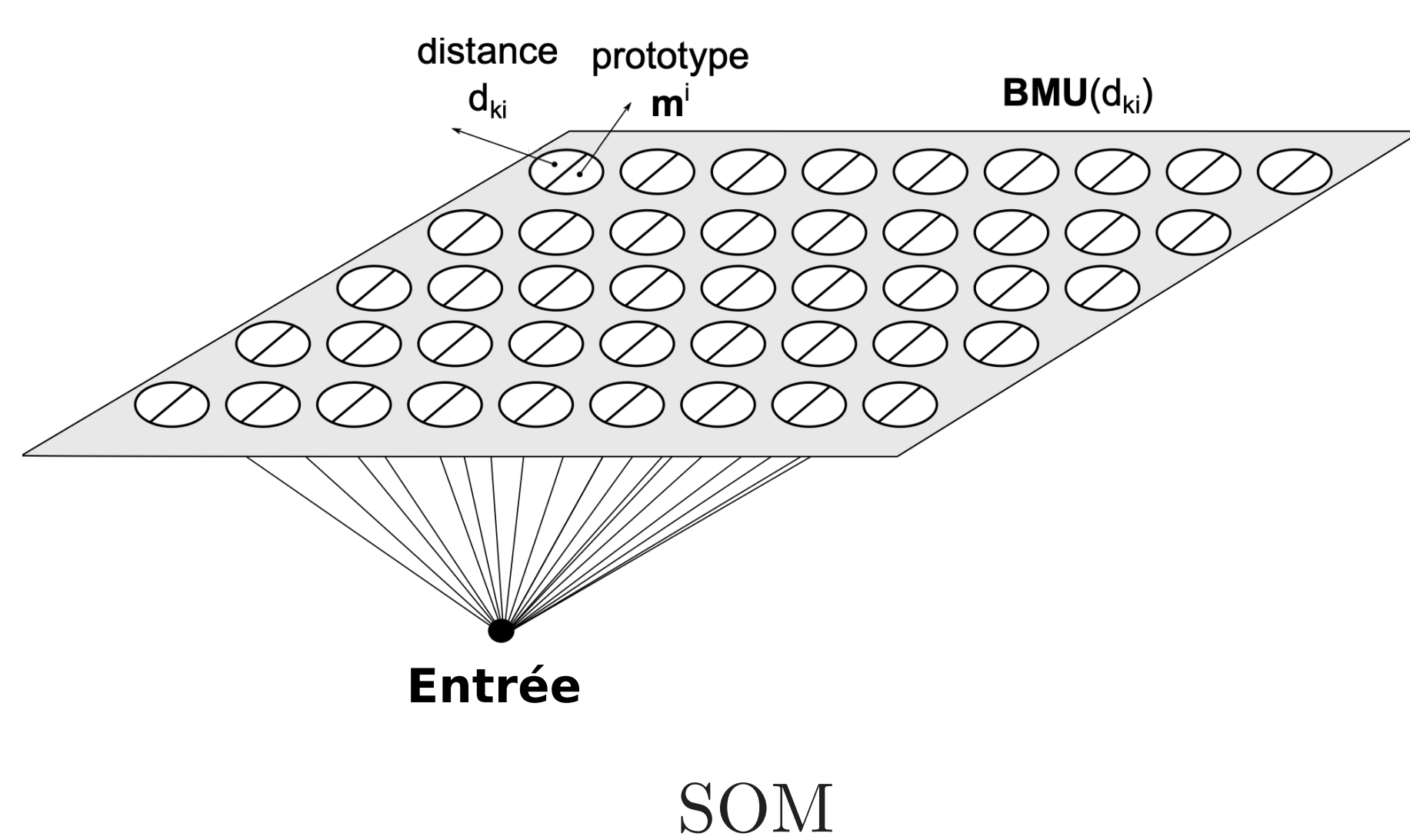
On fait varier un ensemble de paramètres environnementaux pour étudier leurs impacts sur l'erreur d'estimation de  $x, y, \theta$ .

Les paramètres environnementaux considérés sont la vitesse angulaire du plan, la vitesse de l'objet suivi, l'angle de rotation autour de l'axe  $x$  de la caméra 1 et de la caméra 2 et enfin la graine pour la génération du mouvement aléatoire du cube.

## 2. Architecture

L'architecture se décompose en quatre modules :

1. Estimer, pour chaque caméra  $i$ , la vitesse moyenne  $v_i$  entre deux images successives via un algorithme de flux optique.
2. Fournir  $v_i$  comme base d'apprentissage à une carte auto-organisatrice (SOM) [2]. Calculer un coefficient d'allocation de ressource  $c_i$  à partir de la distribution des poids apprise par la SOM.
3. Fournir les images en entrée à des champs neuronaux dynamiques (DNF) et simuler leurs dynamique. Actualiser leurs tailles en fonction du coefficient  $c_i$ . Extraire la position de l'objet suivi de l'activité des DNFs.
4. Estimer, à l'aide de méthodes d'optimisation et d'une fonction de coût basée sur un modèle de vision, la position  $x, y$  de l'objet suivi et l'angle  $\theta$ .



## 4. Conclusion

L'intérêt d'une allocation dynamique de neurones s'adaptant aux fluctuations environnementales a été testé.

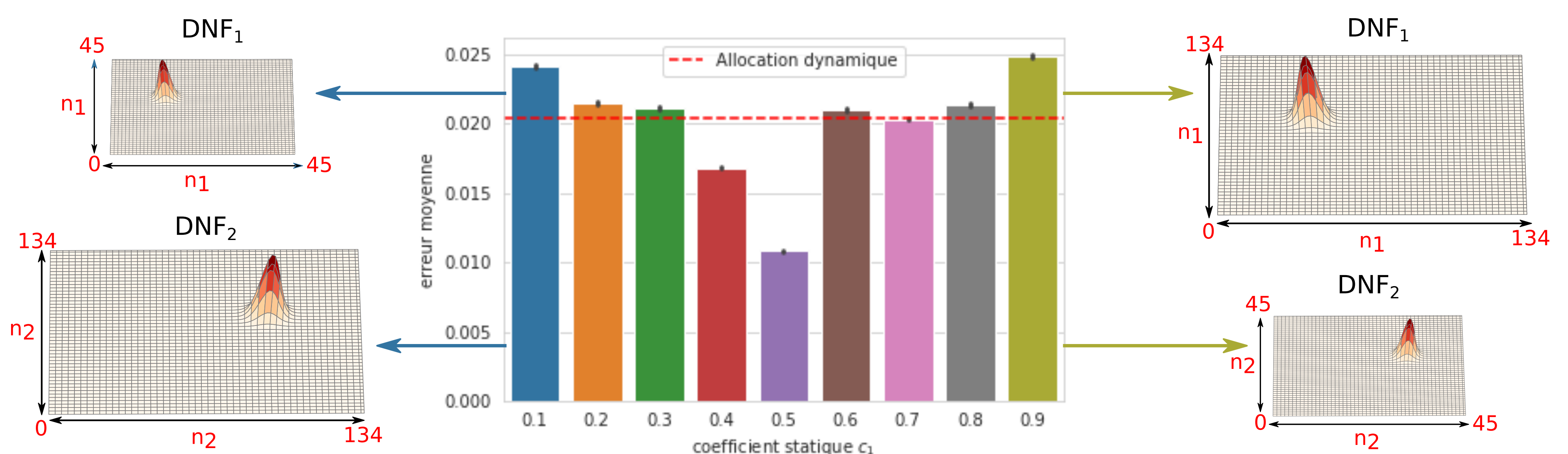
1. Résultats donnés par l'allocation dynamique dans la moyenne vis-à-vis d'une allocation statique.
2. Cela est dû à la nature de l'application, i.e. changer le nombre de neurones impacte la résolution mais pas la dynamique.
3. Une classe d'application plus adaptée serait la compression d'information.

## 5. Références

- [1] Miramond, Benoît: *SOMA*. <http://sites.unice.fr/site/bmiramond/SOMA/>, novembre 2019.
- [2] Kohonen, Teuvo: *Essentials of the self-organizing map*. *Neural Networks*, 37 :52–65, janvier 2013.

## 3. Impact de l'allocation dynamique de neurones

Les performances sont testées sur un ensemble de vidéos. Chaque vidéo correspond à une combinaison de paramètres environnementaux pour laquelle 3000 images ont été générées. Les performances moyennes de l'allocation dynamique de ressources sont comparées avec une allocation statique.



Dans le cas d'une allocation dynamique, la SOM est entraînée pour chaque vidéo en tirant aléatoirement les vecteurs d'entrée. Lorsque la phase d'apprentissage est terminée, les vecteurs d'entrée sont fournis chronologiquement à la SOM. Pour chaque vecteur d'entrée, la SOM donne en sortie un coefficient d'allocation de ressources par DNF.

Dans le cas d'une allocation statique, il n'y a pas d'apprentissage, i.e. les coefficients d'allocation de ressources par DNF sont fixés.