Plenoptic Cameras in Real-Time Robotics

Fengchun Dong · Sio-Hoi Ieng · Xavier Savatier · Ralph Etienne-Cummings · Ryad Benosman

Abstract Real-time vision-based navigation is a difficult task largely due to the limited optical properties of single cameras that are usually mounted on robots. Multiple camera systems such as polydioptric sensors provide more efficient and precise solutions for autonomous navigation. They are particularly suitable for motion estimation because they allow one to formulate a linear optimization. These sensors capture the visual information in a more complete form called the plenoptic function that encodes the spatial and temporal light radiance of the scene. The polydioptric sensors are rarely used in robotics because they are usually thought to increase the amount of data produced and require more computational power. This paper shows that these cameras provide more accurate estimation results in mobile robotics navigation if properly designed. It also shows that a plenoptic vision sensor with a resolution ranging from $3 \times 3$ to $40 \times 30$ pixels camera, provides higher accuracy than a mono-Slam running on a $320 \times 240$ pixels camera. The paper also gives a complete scheme to design usable real-time plenoptic cameras for mobile robotics applications by establishing the link between velocity, resolution and motion estimation accuracy. Finally, experiments on a mobile robot are shown allowing for comparison between optimal plenoptic visual sensors and single high resolution cameras. The estimation with the plenoptic sensor is more accurate than a monocular high definition camera with a processing time 100 times lower.

Fengchun Dong
Xavier Savatier
Instrumentation, IT and Systems Department IRSEEM, Rouen, France

Corresponding authors:
Siohoi Ieng
Ryad Benosman
UMC Univ Paris 06, UMR 7210, UMR_5968, The Vision Institut, 75005 Paris, France
email: sio-hoi.ieng@upmc.fr; ryad.benosman@upmc.fr

Ralph-Etienne Cummings
Computational Sensory-Motor System Lab, The John Hopkins University, Baltimore, United Sates
1 Introduction

Efficient vision-based navigation in mobile autonomous robotics is usually a difficult task to achieve due to the real-time requirements and limited computational power available. Current approaches rely on the use of one or two perspective cameras simultaneously. Adding more camera is generally incompatible with real-time robotic as more data should be processed (Rivas Lopez and Tyrsa (2008)). Standard perspective cameras capture only one ray from each point in space. Motion is generally estimated from the periodic measurements of light as the camera moves in the scene. This estimation is generally non linear and difficult to solve because it is ill-posed (Neumann et al. (2002)). Monocular vision systems also have limitations due to their narrow field of view that introduces ambiguities in distinguishing small rotations from translations or in identifying the exact scale factor.

These problems can be solved by increasing the number of cameras so that larger portions of the scene are covered. However this option does not appear optimal for robotics as it introduces an increase of the amount of data to acquire, transfer and process, thus requiring more computational power. Multiple camera systems acquire multiple rays from the same scene point allowing the estimation of motion to be linear. A more complete light model introduced in (Adelson and Bergen (1991)) is also used for a more accurate motion estimations. This light model is expressed by the plenoptic function which is a 6D mapping (if light wavelength is neglected) used to represent time-varying light rays. An imaging device captures only a small subset of the plenoptic function. If a plenoptic camera captures the plenoptic function at few locations in space, it can then be seen as a set of classic perspective cameras. These sensors are generally referred to as plenoptic cameras (Adelson and Wang (1992)).

This principle originates from a photographic technique called “Integral Photography” described in (Okoshi (1976)). A plenoptic camera is composed of several cameras set close to each other, it can then simultaneously capture a large number of images. Recent technological advances have allowed the production of several plenoptic related cameras, namely, Stanford’s multi-camera array (Wilburn (2004)), the Panoptic camera (Raboud (2009)) and micro lens array cameras (Ng et al. (2005); Raskar et al. (2008); Georgiev and Lumsdaine (2010)). Handheld plenoptic cameras are also available in the market by companies such as Raytrix or Lytro. Other small vision-sensor chips are also developed for applications in which thickness is an issue (Brady and Morrison (2000); Tanida J. Yamada (2002); Duparré et al. (2004); Völkel and Duparré (2008)).

Evolution has selected multiple camera systems when it comes to processing complex visual information with the lowest power consumption. Insects with limited cognitive abilities rely on multiple visual sensors that prove to be particularly efficient for performances that current robotics struggle to attain (Franceschini (2008)). There is a wide variety of compound eyes, in shapes and in size, allowing insects to perform complex manoeuvres such as reactive obstacle avoidance (Land and Collett (1974)), fast prey detection (Olberg et al. (2000)), target localization and tracking (Forster (1979); Jackson and R. (2002)), and accurate vision based odometry (Srinivasan et al. (2000)) at a very low power cost. Compound eyes also allow a wide coverage of scenes and can provide in some cases, an omnidirectional field of view. Insect brains seem capable of processing the amount of information mainly because of two factors. First, the compound eyes have a more optimal coverage of the scene and second, they allow for measurement of light from several viewpoints and directions.
The high resolution needs are consequently reduced as the number of necessary pixels decreases. For example, while the number of photosensitive elements, the ommatidia (or unit eyes), approximates 8000 in bees, it stands around 30000 in dragonflies (Tinbergen (1980)).

Works on motion estimation with compound eyes first focused on Elementary Motion Detectors (EMD) and then transposed to the plenoptic cameras. EMD are based on models of insect motion detection (Hassenstein and Reichardt (1956); Borst and Egelhaaf (1993); VanSanten and Sperling (1984)), with an important property being that motion detection is sensitive to image contrast (Borst and Egelhaaf (1993)). As such, the amplitude of motion detected will be greater where higher contrast exists given the same underlying motion. The EMD are therefore incapable of providing precise visual motion estimates unless an automatic gain control in the ommatidia is implemented. The EMD are usually used in reflexive obstacle avoidance and stabilization in flying robots where only discrete motion adjustments are required. Some recent examples include (Iida and Lambrinos (2000); Ruffier and Franceschini (2003); Liu and Usseglio-Viretta (2001); Wei Chung et al. (2003)). However, given their lack of precision, EMD are not suitable for general navigation, particularly tasks requiring fine motion control.

The applications of plenoptic cameras in real robotics tasks are still scarce. This is probably due to the fact that currently available plenoptic sensors are built for computational imaging tasks. These sensors aim at studying the principle of integral imaging that allows for image refocusing, realistic image rendering, building high dynamic range images and achieving 3D imaging (Camahort and Fussell (1999); Levoy and Hanrahan (1996); Gortler and Grzeszczuk (1996); Adelson and Wang (1992)).

A recent work close to the one presented in this paper has been published in Dansereau et al. (2011) where the plenoptic function is used to achieve realtime navigation. The approach is however fundamentally different from the presented work. In Dansereau et al. (2011), the authors apply the plenoptic function to a set of cameras with fixed parameters. They introduce three distinct closed form solutions to extract the motions parameters from the plenoptic function. Our effort focuses on studying how to reduce data to allow real-time processing without altering the accuracy of the visual odometer by optimizing the sensor’s parameters. The presented work paper inquires into the problem of defining the optimal plenoptic camera for a robotics task. It compares the use of a single high resolution camera versus a lower resolution plenoptic sensor. At the core of the question is the problem of visual navigation and motion estimation: what plenoptic camera should be built for a specific robotics navigation task according to a given range of speed of the robot? What is the lowest possible resolution of a plenoptic camera needed to subsequently facilitate the motion estimation problem in the best possible way and perform as well or even better than a single high resolution camera?

We will show that there is an optimal set of parameters allowing a tradeoff between resolution, field of view and accuracy. The optimal plenoptic camera reduces drastically the unnecessary data with minimal accuracy loss. As will be shown, high resolution sensors are often less efficient in most navigation tasks, performing worse than a carefully designed plenoptic camera with a much lower resolution.
2 Motion and structure from the plenoptic function

2.1 Motion from the plenoptic function

The plenoptic function is a more complete physics model of light that describes an image from a bundle of sampled rays where each ray is characterized by the position and orientation from which it is seen, at a given time \( t \). Such a light model defines the structure of the visual space, noted \( L(x, r, t) \), with:

- \( L \) as the radiance,
- \( x = (x, y, z)^T \in \mathbb{R}^3 \) as the viewing position,
- \( r = (\theta, \phi)^T \in [0, 2\pi]^2 \) as the direction from which \( L \) is seen,
- \( t \in \mathbb{R}^+ \) as the time.

The plenoptic-based motion estimation is applied with the formalism established in (Neumann et al. (2002, 2004)). The vision sensor is moving through the scene and each of its cameras at \( x \) samples the plenoptic function from \( r \), at \( t \). The egomotion is rigid i.e. given by a rotation matrix \( R \) and a translation vector \( T \).

Estimating the motion parameters from the plenoptic function assumes the smoothness of the signal, which allows a Taylor’s first order expansion. A second hypothesis is the constant scene illumination overtime. If this is fulfilled, the plenoptic function satisfies the photo-consistency constraint which is similar to the one used in optical flow computation (Neumann (2004)):

\[
- L_t = (\nabla_x L)^T \frac{dx}{dt} + (\nabla_r L)^T \frac{dr}{dt}.
\]

where \( \nabla_x L \) and \( \nabla_r L \) are the spatial gradients of \( L \) and \( L_t \) its partial time derivative \( \frac{\partial L}{\partial t} \).

If \( \omega = (\omega_x, \omega_y, \omega_z)^T \) and \( q = (q_x, q_y, q_z)^T \) are respectively the angular velocity and the instantaneous translation, then Eq. 1 becomes:

\[
- L_t = (\nabla_x L)^T q + (x \times \nabla_x L + r \times \nabla_r L)^T w,
\]

where \( \times \) is the cross product. This equality is referred to as “differential plenoptic motion constraint” which is a linear, scene-independent constraint in the motion parameters and the plenoptic partial derivatives. The mixture of cartesian and spherical coordinates \((x, r)\) in Eq. 2 is inconvenient to manipulate. This equation is replaced by a more flexible but equivalent form derived from the two-planes parametrization (Levoy and Hanrahan (1996)). Eq. 2 can then be rewritten by computing the Jacobian between the original and the two-planes parametrizations, and then simplified into the matricial form:

\[
- L_t = L_z \begin{bmatrix} L_y & L_u & L_v \end{bmatrix} M \begin{bmatrix} q \\ \omega \end{bmatrix},
\]

with

\[
M = \begin{pmatrix}
1 & 0 & -\frac{u}{f} & 0 & 0 & 0 \\
0 & 1 & -\frac{v}{f} & 0 & 0 & 0 \\
-\frac{u}{f} & -\frac{v}{f} & \frac{1}{f} - Z_{\pi} & \frac{u x}{f} & \frac{v y}{f} & \frac{w z}{f} \\
-\frac{u}{f} & -\frac{v}{f} & \frac{1}{f} & \frac{u^2}{f} & \frac{v f}{f} & \frac{u x}{f} \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix},
\]

the transformation matrix for the translation and rotation.

The meaning of each variable in \((x, y, u, v, f, Z_{\pi})\) is detailed in (Neumann (2004)). An over-determined linear system can be built with the contribution of each camera and solved for.
the rigid motion parameters with standard least square techniques.

The mechanism explaining why a plenoptic camera is more suitable for navigation task is shown in Fig. 1. When moving through the scene, a conventional camera can truly satisfy the photo-consistency expressed by Eq. 1 only for small motions while the plenoptic camera can handle much larger displacements. For such motions, a single ray can actually be captured by another camera unit in the plenoptic device, while for a single camera, this property does not hold anymore.

![Fig. 1: (a) A monocular camera and (b) a plenoptic system observing the same scene while undergoing the identical rigid motion. In (a), the photo-consistency is true for small motions or if the scene reflects light isotropically since the camera is observing two different rays (dashed arrows) from its two positions. In (b), the photo-consistency is true because a same ray can be captured by two different cameras (e.g. the top dashed arrow captured by the sensor at $t$ and $t + 1$).](image)

2.2 Multi-scale plenoptic function

Cameras’ motions across the scene span from slow, rectilinear translations to complex and non coplanar trajectories. The resulting velocity field is a complex mixture of motions which hardly satisfies the constant illumination hypothesis, especially for large displacements. Correct motion parameters would sometimes be impossible to estimate. One classic solution is to apply a multi-scale representation of the signal and to perform a multi-layered motion estimation, starting from the coarsest level that underlines larger motions and erases smaller ones. The results are propagated to the next finer level to contribute to a refined motion estimation. Classical application of the multiscale technique consists in building a gaussian pyramid on the image space.

In our case, a pyramid structure for a vector field $l$ that samples the plenoptic function data is built and defined as follows: $l = (x, y, u, v, L_x, L_y, L_u, L_v, L_t)^T$.  

$$l : \mathbb{R}^5 \to \mathbb{R}^{10}, \quad (x, y, u, v, L_t) \mapsto l = (x, y, u, v, L_x, L_y, L_u, L_v, L_t)^T,$$

where $L_x, L_y, \ldots$ are the partial derivatives of the plenoptic function with respect to each variable.

The multi-scale strategy is related to the scale space theory (Lindeberg (1994)) which is a
formal theory that maps any vector field to a one-parameter family of smoothed vector fields. In the case of \( l \), the multi-scale space represents a family of subsampled signals \( l(x, r, t, k) \) smoothed by gaussians of variance \( k \), with \( k \) being also the scale parameter.

**Algorithm 1** Multi-scale motion estimation.

1. **Inputs**: Motion parameters estimated at level \( k + 1 \), the gaussian and plenoptic pyramids.
2. **Outputs**: Motion parameters \((q, \omega)^T\).
3. **for each level** \( k \) **do**
4.   **Build Eq.** (3) **with current level data and solve for the motion parameters**: \((\tilde{q}, \tilde{\omega})^T\).
5.   **Compute the plenoptic function and its derivatives with current level data, then update odd rows and columns of the plenoptic pyramid. Even rows and columns are updated with level \( k + 1 \) of the plenoptic pyramid.**
6.   **For each element at** \((x, y)^T\) **of the pyramid, update it with a weighted mean value of the plenoptic vector over a 3 \( \times \) 3 neighborhood (including the data of current level \( k \) and the data of previous level \( k + 1 \)).**
7.   **Build a new Eq.** (3) **with the updated pyramid data and solve for motion parameters**: \((q, \omega)^T\).
8.   **if** \( ||(q, \omega)^T - (\tilde{q}, \tilde{\omega})^T|| < \text{threshold} \) **then**
9.     **exit and return the motion parameters**
10. **end if**
11. **end for**

The motion estimation is performed by iterative propagations from the lowest resolution (highest level) to higher ones (lower level) with the possibility to stops and returns the results if the motion estimation satisfies some prefixed conditions. The pyramids are assumed n-level high, with the highest level storing the lowest resolution data. The iterative motion estimation operates according to algorithm 1. The threshold mentioned in line 8 is determined experimentaly using a set of trajectories of the robot, it is taken as the mean value giving the lowest estimation error. This value is equal to \( 5 \cdot 10^{-2} \).

### 2.3 Optimal scale factors

The scale space theory is widely used in signal and image processing. Lindeberg (1994) gives a detailed formalization and analysis of the scale space representation for 2D image signals. In this section, we extend the multiscale representation to the plenoptic function i.e. instead of building a pyramidal set of 2D images, we build a pyramid of 10-dimensions function.

The multiscale approach is motivated by the ability to deal with a wide range of natural scenes. The sensor motion estimation from the image is largely conditioned by image structures’ size and depth. Conversely, the sensor motion’s velocity and direction are directly impacting the resulting image signal. To reduce noise that limits the estimation quality, one should find the best scale of the signal representation that captures the most accurately possible, the sensor’s egomotion. Finding a unique glabal optimal scale assumes that motion components (i.e. instantaneous translation \( q \) and instantaneous rotation \( \omega \)) are affected in a similar way. However, according to Eq. 2, we can see that \( q \) depends only on the viewing positions while \( \omega \) depends on both the viewing positions and rays direction. This underlines that translations are mainly captured by \( \nabla_x L \) while the instantaneous rotation are sensitive to more local changes.
According to this observation, we stipulate that both terms are not influenced by the scale factor in a similar way and each component should be optimized with its own scale factor. The best scale is given by the one producing the largest signature operators response. For example, in edges detection, the trace or the determinant of the Hessian matrix should be maximized while in texture characterization, the hessian matrix is replaced by the second moment matrix. Fig. 2 shows the response of the second moment matrix $\mu$ computed at the center of each image:

$$
\mu = \begin{pmatrix}
L_x^2 & L_x L_y \\
L_y L_x & L_y^2
\end{pmatrix}.
$$

More precisely, $\det \mu$ is computed for each scale and the one maximizing its value gives the optimal scale for texture characterization. This shape signature is actually one of those used in Lindeberg (1994) to set an automatic scale detection. To underline that the scale has different influence on the $\nabla_x L$, $\nabla_y L$ (i.e. the inter camera derivatives) and the $\nabla_u L$, $\nabla_v L$ (i.e. intra camera derivatives), we computed the determinant of $\mu$ for each set of gradients (normalized by the maximum value). As we can see, in one case the maximum responses is reached at scale $(160 \times 120)$ for the intra camera derivatives and scale $(10 \times 8)$ for the inter camera derivatives. In the second case these values are respectively $(40 \times 30)$ and $(10 \times 8)$. Hence, the scale optimizing the estimation of $q$ and the one optimizing $\omega$ are different.

This observation is a major issue as it requires to redesign Algorithm 1 to optimize separately the estimation of $q$ and $\omega$. One way to do it is to apply Algorithm 1 to find the optimal scale that stabilizes $q$, then fixes $q$ with the estimated value and re-apply Algorithm 1 to optimize $\omega$.

### 3 Experiments

#### 3.1 Setup

The vision sensor used for the experiments is built out of a set of nine perspective cameras arranged in a $3 \times 3$ array, reproducing the functioning of a compound eye (see Fig. 3). These cameras are configured to work synchronously with the same frame rate. A rigid mount was also designed to ensure the stability of the array structure spacing cameras by $5cm$ both horizontally and vertically. The system is then calibrated with the multi-camera calibration method described in (Svoboda et al. 2005), giving the relative pose of each camera. Their synchronization is achieved by sending a triggering signal to start the acquisition from the unique computer which also collects all the incoming data streams. The image maximal resolution is set to $320 \times 240$ pixels as a tradeoff between signal resolution and available bandwidth allowed by the connexion to the computer. Finally, the entire system is embedded on a mobile Pioneer platform to record data while the latter executes trajectories at various speeds up to $1.8m \cdot s^{-1}$. Higher translational speeds which are beyond the platform’s ability are approximated by changing the cameras’ frame rate. The estimation results at these speeds should be considered as the upper bounds of the odometer’s accuracy.

In a fast acquisition process, the acquired images will suffer from motion blur as the platform’s velocity increases. Results should however compare as for speeds exceeding $2m \cdot s^{-1}$ higher frame rates are usually used.

A Kinect sensor is used to ensure ground truth and assess the motion estimation performances. It is calibrated using (Burrus 2011) so that each pixel in the light intensity image is mapped to the depth image. Hence, the trajectory of the robot moving in the scene can be estimated directly. Tracked motions are also constrained within the Kinect’s range of accu-
Fig. 2: Mean value of the determinant of the second moment matrix $\mu$ computed at the center of each image. The determinant is used as scale-space signature. The plain curves are the responses obtained for $L_u$, $L_v$ (intra camera gradient) while the dashed curves are the ones for $L_x$, $L_y$ (inter camera gradient). In each case, the maximums are not the same as we conjectured: scales $160 \times 120$ and $10 \times 8$ for the top and scales $40 \times 30$ and $10 \times 8$ for the bottom.

3.1.1 Results

Five types of motions are executed by the mobile robot with the plenoptic system embedded on it, and for each motion a total of six trials are performed. Fig. 4 shows a set of circular trajectories corresponding to six trials. Each of them are estimated with the plenoptic visual odometry for a total of 100 samples, only a fraction of them are plotted as markers for readability purpose.

Fig. 5 shows the histogram of the translational errors for the all the mentioned motions. These errors are represented as absolute errors with a mean value equals to 0.07m. All trajectories have an average length of 10m. We have also applied the monocular algorithm designed in (Civera et al. (2010)), that has been largely tested and used for robotic navigation, on the same set of data. The estimated motion parameters obtained using either technique are compared with the same ground truth and the normalized errors are shown in Fig. 6. Finally and because it is usually claimed being the gold standard in SFM problems,
Fig. 3: Experiment setup: a 3 × 3 camera array on a mobile robot monitored by a Kinect sensor. The camera array captures nine video sequences at 20 fps while the Kinect provides the ground truth of its motion.

The bundle adjustment is known for providing reliable and stable solutions. We then also apply to the data the monocular SLAM algorithm presented in Strasdat et al. (2010). As one can expect, the error curve is more stable, the algorithm even produces the best performances at the beginning of the motion and is slightly less performant than the plenoptic method as the distance increase.

The relative errors are normalized by the traveled distance. The plenoptic algorithm is performing better than the monocular algorithm as the error is always lower. The bundle adjustment algorithm proves to work efficiently by giving the lowest mean error at short distances. As distance increases, the bundle adjustment algorithm error is slightly above the plenoptic estimation’s one. The mean error is around 2.85% for the plenoptic estimation while the Civera’s monocular SLAM provides a mean error of 6%. Strasdat’s algorithm is giving a lower mean error of 2.5%. The decreasing behavior of the curves underlines the stability of the estimations techniques: the errors are decreasing as the lengths of the trajectories increase, meaning that the errors are bounded all the time.

The monocular algorithm performances drop as the motion speed increases. Above the speed of 1 m · s\(^{-1}\), the monocular algorithm often fails to estimate the motion. One main reason of this failure is the difficulty to set the initial guess of the Kalman filter used in the algorithm. Without prior knowledge of the motion, the algorithm diverges at high speed if the initial guess is chosen loosely. Another reason of the poor performances at high speed are due to motion blur which increases with the velocity and causes feature tracking difficult
to fail when only one camera is used. Cameras array used in conjunction with the plenoptic function is a featureless technique, it is more robust to large displacements as explained in section 2.1.

3.2 Optimal setup

Motion estimation accuracy varies according to the motion speed, sensor resolution, and field of view. Although there is no simple way to express analytically the estimation error with respect to those variables, optimal parameters must be identified in order to produce the most accurate motion estimation. As discussed in previous sections, the multi-scale approach aims to recover the motions accurately while avoiding to process images at unnecessarily high resolution levels. However, this is still a computationally expensive process that
Fig. 5: Histogram of the translational absolute errors estimated by the plenoptic algorithm, for the six trials of the five motions. The average length of the trajectories is 10m and the mean estimation error is equal to 0.07m.

Fig. 6: Normalized translational errors with respect to the traveled distance. The triangles show the estimation results from the monocular algorithm, the squares show the plenoptic ones and finally the circles represent the results from the SLAM algorithm incorporating the bundle adjustment. The errors are computed for 5 types of trajectories, each repeated 6 times.

prevents realtime applications. By identifying optimal parameters for certain motions, it is possible to reduce the search domain and more important decrease the processing time.

3.2.1 Impact of resolution

The influence of the resolution is examined in this section. The estimation error \( f \) is expressed as a function of the scale factor and the velocity. The optimal parameters are deter-
mined by minimizing $f$ with a standard gradient descent:

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$\mathbf{X} \rightarrow f(\mathbf{X}),$$

where

$$f(\mathbf{X}) = \frac{||(\mathbf{q}(\mathbf{X}), \mathbf{\omega}(\mathbf{X}))^T - (\tilde{\mathbf{q}}(\mathbf{X}), \tilde{\mathbf{\omega}}(\mathbf{X}))^T||^2}{||(\mathbf{q}(\mathbf{X}), \mathbf{\omega}(\mathbf{X}))^T||^2},$$

and $\mathbf{X} = (k, v)^T$ is the vector of the scale factor and the velocity. The scale factor is the same one that is introduced as the scale space parameter. Fig. 7 shows the error function established for velocity ranging from 0.2 to 5 m $\cdot$ s$^{-1}$ and resolution from (320 $\times$ 240) to (5 $\times$ 4) pixels. A local minimum of $f(\mathbf{X})$ is found at $\mathbf{X} = (4.46, 1.32$ m $\cdot$ s$^{-1})$, i.e. at a resolution of 29 $\times$ 22. This minimum is also confirmed by the experimental results. The algorithm selects frequently the level 20 $\times$ 15 when the speed $\in [0.9, 2]$ m $\cdot$ s$^{-1}$. The motion estimation accuracy is usually expected to be an increasing function of the

![Fig. 7: Estimation errors with respect to the signal resolutions and motion velocities. The plenoptic sensor is particularly adapted for speeds from 1 to 2 m $\cdot$ s$^{-1}$ for resolutions bewtween 40 $\times$ 30 and 20 $\times$ 15 pixels.](image-url)

sensor resolution. However experiments show surprising results where lower resolutions yield superior results than higher ones. This counter-intuitive result can be explained by the fact that given the robot’s speed and the scene content, the ego motion estimations are often biased by local minor motions or lighting changes. Reducing the resolution filters out these perturbations. This process has a limit as shown in Fig. 7. A resolution that is too low tends to provide highly imprecise estimations. Hence, the estimation error has the shape depicted in Fig. 7, with a local minimum corresponding to the optimal parameters. We may then wonder if there exists a methodology to predict and exploit this relationship, rather than using the brute-force search presented? Unfortunately, this behavior is largely dependent of the scene contents for which we usually do not have control on. It is then appears difficult to predict an a priori shape to the estimation error. However for large classes of scene a “mean” optimal parameter set could be probably estimated.
3.2.2 Impact of the field of view.

The field of view is one of the critical parameters for motion estimation problems. It is known that visual sensors with narrow field can not distinguish translations from rotations (the aperture problem Hildreth (1984)). Increasing the field of view can however significantly decrease these ambiguities. In this experiment we start by studying its influence by iteratively cropping the original images from their maximal resolution of $320 \times 240$ pixels to artificially decrease resolution. Each camera is mounted with identical lenses providing a $35^\circ$ field of view both vertically and horizontally. Fig. 8 shows the motion estimation results for arbitrary translations with different values of the field of view expressed as ratios of the maximal value (i.e. $35^\circ$). From the error curves, one can see that the motion estima-

![Error Curves](image)

Fig. 8: Normalized translation errors plotted with a logarithmic scale to make the errors at slow motions ($< 1\, m.s^{-1}$) visible. Each curve has a fixed field of view ratio value.

...tion is the most accurate when the sensor is working with its maximum field of view. The lower the field, the faster the estimation performance decreases with estimation errors higher than 10%. For too small values of the field of view the algorithm simply fails because of the lack of sufficient overlapping areas. It also induces a subsequent reduction in the visual cues required for the photo consistency.

The nine used cameras have fixed focal length lenses, it was then not possible to study the impact of higher values of the field of view. In order to provide upper bound results, we generated data from artificial scenes filled with textured spheres randomly distributed. In this second experiment the field of view is studied for a range of 10 to $150^\circ$, with cameras’ spacing ranging from 5 to 30 cm. The remaining parameters are set to the same values as the ones of the developed plenoptic sensor. The virtual camera velocity is set to $1\, m \cdot s^{-1}$. The multi-resolution algorithm is applied to estimate motion providing mean errors shown in Fig. 9.

A mean error of 12% is achieved for a field of view of $35^\circ$ degrees and cameras spacing of 5 cm. This is also consistent with the results presented on real data in section 3.1.1. Error curve reported in Neumann et al. (2002) provide similar conclusions, the error decreases as
the field of view becomes larger. It is also less sensitive to the cameras spacing at large values of the field of view. The multi-resolution method reduces the amount of errors by 10%. This can be explained by the optimization process that selects the most suitable resolution level given the speed of $1 \text{m} \cdot \text{s}^{-1}$. According to the error curve, one can expect with this approach to go below 5% of error with a $65^\circ$ field of view.

We intentionally discarded the study of spatial arrangement of cameras in order to optimize the field of view, this goes beyond the scope of this paper, as the practical experiments and comparisons would then have to use omnidirectional sensors for comparisons, which is a complete other topic.

### 3.2.3 Processing time

The plenoptic motion estimation algorithm has been tested on a Core 2 Duo, running at 2.40Ghz, the algorithm is implemented in Matlab without specific optimization. The motion estimation is performed independently for each resolution level, starting from $5 \times 4$ to $320 \times 240$ pixels. Regardless to the estimations’ accuracy, the mean processing times at each step (i.e. pair of images) are measured for all the images of each sequence and shown in Table 1. The mean processing time is also measured for both the monocular technique that also relies on a matlab implementation and the multi-scale plenoptic algorithm which is supposed to provide the optimal results with the shortest processing time. The results are shown in the two last rows of Table 1.
Table 1: Mean processing times for the single resolution plenoptic motion estimation algorithm at each resolution level compared to the multi-scale and monocular algorithms.

These results show that by selecting the optimized resolution (here 20 × 15), accurate motion estimations can be achieved. Compared to the multi-scale technique, the processing time is also shorter by a factor 100. Beyond the benefit of a drastic reduction in processing time, we also gain accuracy in the motion estimation. In our experiments, the plenoptic algorithm works remarkably well at the resolutions of 40 × 30 and 20 × 15 pixels, with a mean processing time of 0.03 s, while the monocular algorithm works at 1.32 s and the multi-scale at 3.15 s.

3.2.4 Proposed setup

The impact of image resolution and field of view were analyzed in previous sections. These tests help to determine the optimal setup to produce correct estimation at an optimized computational time. The minimization of the error function in Eq. 5 allows to find the adequate parameters \( \mathbf{X} = \arg\min f(\mathbf{X}) \) where \( \mathbf{X} \) is the vector of the sensor parameters (scale level and field of view) and the motion velocity. The motion velocity is not a parameter of the sensor, however it is of great importance to the vision system. The velocity range is in principle known from the specification of the robot’s architecture. This brings relevant information to properly initialize the motion estimation process.

Table 2: Optimal setups according to motion speeds. Three domains of velocity define the optimal estimation parameters which allows for accurate estimation and short processing time.

Table 2 can be seen as a chart to help designing adapted vision sensors for a specific robotic task. It summarizes the setup producing the lowest mean estimation error according to the motion speed. The presented work dealt with sets of similar cameras (i.e. similar
intrinsic parameters, regular spacing) as the paper initial goal is the study of the computation properties of insects’ compound eyes that generally rely on similar visual units. More general configuration would be interesting to study, we believe they are linked to higher "cognitive" tasks such as predation, reproduction,....

The error function is minimized over the variable $X$ with a gradient descent to find the local minimum for each speed interval: $< 0.7 \text{m} \cdot \text{s}^{-1}$, $[0.7, 2] \text{m} \cdot \text{s}^{-1}$ and $[2, 2.6] \text{m} \cdot \text{s}^{-1}$. For a standard wheeled robot (e.g. Pioneer, with a maximum speed of $1.8 \text{m} \cdot \text{s}^{-1}$), the optimal choice will be a $40 \times 30$ pixel plenoptic camera, it produces the best balance between processing speed and accuracy. In the case of fast robots such as flying UAV $20 \times 15$ pixel plenoptic camera with a wide field of view can significantly reduce the processing time while maintaining an error estimation of less than $12\%$. For higher speed motions, ($v > 2.6 \text{m} \cdot \text{s}^{-1}$), there seems to be no real optimal set of parameters as the sensor estimation errors are always higher than $20\%$.

4 Conclusion

This paper presents an in-depth exploration of the plenoptic approach of a vision-based navigation. The limited fov of a perspective camera was pointed out as one of the main causes of the poor performance of many widespread vision algorithms but several factors are also examined and are shown to contribute to the motion estimation quality: the sensor’s resolution, the motion velocities of the sensor while moving, etc. Experiments performed in real conditions with a mobile robot show the plenoptic function approach often outperforming the state-of-the-art vision algorithms. The coarse to fine scheme of the approach produces a bottleneck in the motion estimation process, but it is only exploited to identify the ideal parameters allowing to use the vision system at its highest accuracy with minimal data payload. With the adequate parameters for the cameras, we show that a plenoptic vision system is not only a more accurate sensor for motion recovery from the images, but it also has the ability to handle larger motions. Most interestingly, such performances are achieved with an extremely low processing power requirement, thus making it perfectly suitable for real-time robotics.

The plenoptic vision sensor is presented as the optimal technological solution to the problem of vision-based motion estimation, raising several promising expectations. While classical vision-based navigation techniques struggle to perform as efficiently as living organisms, despite the piling up of complex processing techniques, the plenoptic formulation offers an interesting alternative to solving the problem. This work highlights the irrelevance of ever increasing resolution cameras to achieve accurate motion estimation and reinstate the benefit of using a plenoptic vision system in robotic applications. With properly defined optical parameters, it is possible to build a reactive and accurate vision navigation system requiring reasonable computation resources.

5 Discussion

This paper shows a surprising result, that a set of low resolution cameras can be more efficient for navigation than a high resolution visual sensor. This is can be easily explained by the fact that multiple viewpoint acquisition maximizes the signal-to-noise ratio, accurate
conclusions can then be drawn. If the individual measurements of a plenoptic camera have a lot of variability it becomes difficult to determine the useful information needed to perform the task. Measuring a single item or event more than once maximizes the accuracy of results. More measurements of a single event lead indeed to greater confidence in calculating an accurate average estimation.

Another important conclusion of the paper relates to the low amount of data to process. We often think of camera-like eyes when dealing with perception, but nature contains several types of eyes, with an incredible variety of designs as shown in the work of Michael Land (Land and Nilsson (2002)). Compound eyes directly related to this work are found in the insect world. Their visual acuity is around one hundred times less than that of the human eye. The compound eyes are however known to be excellent at detecting motion (Dawkins (1996)). Compound eyes resolution is particularly adapted to the low neural processing ability of insects brain. Each facet of a compound eye encloses one single visual unit (ommatidium) containing around 7 to 11 sensory cells, this corresponds well to the limited processing power of insect brain.

From a technical point of view more adapted cameras could be used. Fig. 1 shows a design of the camera system where each camera is a completely closed unit, composed of a unique lens and image sensor. A more adapted camera could be built to achieve a similar $3 \times 3$ plenoptic camera system using a single image sensor and an array of micro lenses or pin-holes. Current CMOS technology allows video image sensors with as many as 10M pixels over a chip size of 2.5 cm $\times$ 1.4 cm (Thorpe (2011)). Such a camera can easily be organized as a plenoptic camera system when coupled with the appropriate optics. Typically for these types of large format image sensors, the entire image can be readout from the image sensor, or be down-sampled by the readout integrated circuit, as mandated by a feedback loop that sets the required resolution as a function of the estimated visual motion. Hence, a single chip plenoptic camera system using the standard video imaging chips can readily be built and coupled with the visual motion estimation and optimization algorithms discussed above for applications in mobile robotics.

Another major property of insect eyes is their high temporal resolution that was not inquired in this paper. As an object moves across the visual field of a compound eye, ommatidia are progressively turned on and off. Because of the resulting "flicker effect", insects respond far better to moving objects than stationary ones. It is known that honeybees will visit wind-blown flowers more readily than still ones. The compound eyes can detect fast moving objects with high precision, contour detection is sought to be the two most important characteristics of compound eyes. Non-scanned image sensors that use the Address Event Representation (AER) protocol to asynchronously output the pixels are also ideally suited for plenoptic cameras (Culurciello et al. (2003); Lichtsteiner et al. (2006)). In such cameras, the pixels’ addresses, i.e. location in the array, and in some cases the pixel values, are output when the pixels deem that enough light has been integrated by their photo-detectors. These sensors compress information at the level of pixels and have a tremendous temporal resolution of several kHz.

**Funding**

This work was supported by the Council of the Large Research Network in Energy, Electronics and Materials of the french region of Haute Normandie; the ISE-LNA project, implementation of autonomous navigation laboratory.
Acknowledgement

The authors thank gratefully Kathrine Matho for proofreading the paper.

References


