Multiresolution image alignment
– lecture notes –

Pedro M. Q. Aguiar
Institute for Systems and Robotics / IST

April 2008

Abstract

These notes address the problem of aligning, or registering, two images (or image patches). This is done by estimating the motion of the brightness pattern between the two images in a global way, *i.e.*, without any pre-processing step such as selecting and matching salient features. The resultant optimization problem is solved in an efficient way by using a Gauss-Newton method in a multiresolution pyramid.

1 Introduction

These notes address the estimation of the two-dimensional (2D) motion of the brightness pattern in the image plane. This is a crucial step in solving any motion analysis task. For example, the problems of building a mosaic, or panorama, from uncalibrated images, segmenting out moving objects, and inferring three-dimensional (3D) structure from video, require the step of estimating 2D motion to accomplish their higher level goals.

The problem of estimating the 2D motion of the brightness pattern has been widely addressed in the recent past. The cue to estimate the motion of the brightness pattern between two images is the brightness constancy. Since the early days of motion analysis, researchers have noticed that local motion estimation is an ill-posed problem, see for example [1]. In fact, it is easily shown that it is not possible to determine the 2D motion if we are allowed to see only through an infinitesimally small spatial aperture. This limitation, referred to as the *aperture problem*, is usually overcomed by using smoothing assumptions [1]. In these notes, this is done by describing motion in a parametric way. To estimate the motion parameters, we use an efficient approach made popular by Bergen, Anandan, Hanna, and Hingorani [2]. This approach uses a hierarchical Gauss-Newton method, where the derivatives involved are simply computed from the image gradients.

The notes are organized as follows. In section 2 we develop the motion estimation algorithm and discuss the influence of the spatial variability of the brightness pattern on its behavior. In section 3 we study the estimation error. We derive an expression for the variance of the estimation error and discuss how to use this result to measure the accuracy of the estimates of the motion parameters. Sections 4 and 5 particularize the study of sections 2 and 3 to two widely used motion models: translational and affine. Section 6 summarizes.

2 Motion estimation

We consider images as real functions defined on a subset of the real plane. The image space is a set $\{\mathbf{I} : \mathcal{D} \rightarrow \mathbb{R}\}$, where $\mathbf{I}$ is an image, $\mathcal{D}$, a compact subset of the real plane $\mathbb{R}^2$, is the domain of the image, and $\mathbb{R}$ is the range of the image. Images code intensity gray levels[1]. The domain of each image is rectangularly shaped with size fitting the needs of the corresponding image. Although we use a continuous spatial dependence for commodity, in practice the domains are discretized and the images are stored as matrices. We index the entries of each of these matrices by the pixels (x, y) of each image and refer to the value of image $\mathbf{I}$ at pixel (x, y) as $\mathbf{I}(x, y)$. Throughout the text, we refer to the image product of two images $\mathbf{A}$ and $\mathbf{B}$, *i.e.*, the image whose value at pixel (x, y) equals $\mathbf{A}(x, y)\mathbf{B}(x, y)$, as the image $\mathbf{AB}$. Note that this product corresponds to the Hadamard-Schur product, or elementwise product, of the matrices representing images $\mathbf{A}$ and $\mathbf{B}$, not their usual matrix product.

[1] The intensity values of the images are positive, usually coded by a binary word of eight bits. Thus, the intensity values of a gray level image are in the set of integers in the interval $[0, 255]$. For simplicity, we do not take into account the discretization and the saturations, *i.e.*, we consider the intensity values to be real numbers and the gray level images to have range $\mathbb{R}$. The analysis in these notes is easily extended to color images. A color is represented by specifying three intensities, either of the perceptual attributes *brightness*, *hue*, and *saturation*; or of the primary colors *red*, *green*, and *blue*, see, *e.g.*, [3]. The range of a color image is then $\mathbb{R}^3$.

We consider parametric motion models. We represent this kind of motions by specifying parameter vectors. Let N_p be the number of parameters describing the motion. The image obtained by applying the rigid motion coded by the $N_p \times 1$ vector $\mathbf{p}$ to the image $\mathbf{I}$ is denoted by $\mathcal{M}(\mathbf{p})\mathbf{I}$. The image $\mathcal{M}(\mathbf{p})\mathbf{I}$ is also usually called the registration of the image $\mathbf{I}$ according to the parameter motion vector $\mathbf{p}$. The entity represented by $\mathcal{M}(\mathbf{p})$ is seen as a motion operator. In practice, the (x, y) entry of the matrix representing the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ is given by $\mathcal{M}(\mathbf{p})\mathbf{I}(x,y) = \mathbf{I}(f_x(\mathbf{p}; x, y), f_y(\mathbf{p}; x, y))$ where $f_x(\mathbf{p}; x, y)$ and $f_y(\mathbf{p}; x, y)$ represent the coordinate transformation imposed by the 2D rigid motion. We use bilinear interpolation to compute the intensity values at points that fall in between the stored samples of an image. We denote the "inverse" of $\mathbf{p}$ by $\mathbf{p}^{\#}$, *i.e.*, the vector $\mathbf{p}^{\#}$ is such that the registration of the image $\mathcal{M}(\mathbf{p})\mathbf{I}$ according to $\mathbf{p}^{\#}$ is the original image $\mathbf{I}$.

Consider a pair of images $\{\mathbf{I}_1, \mathbf{I}_2\}$. Our goal is the estimation of the motion of the brightness pattern between images $\mathbf{I}_1$ and $\mathbf{I}_2$ in a given region $\mathcal{R}$ of the image plane. The observation model simply captures the intensity constancy, *i.e.*, for pixels (x, y) within region $\mathcal{R}$, we have:

$$\left\{ \begin{array}{l} \mathbf{I}_1(x,y) = \mathbf{S}(x,y) + \mathbf{W}_1(x,y) \\ \mathbf{I}_2(x,y) = \mathcal{M}(\mathbf{p}^{\#})\mathbf{S}(x,y) + \mathbf{W}_2(x,y)\,, \end{array} \right. \tag{1}$$

where $\mathbf{S}$ is the (unknown) scene observed in image $\mathbf{I}_1$; the unknown motion is represented by the vector $\mathbf{p}$ that parameterizes the operator $\mathcal{M}(\mathbf{p}^{\#})$ that "distorts" $\mathbf{S}$ when observed in image $\mathbf{I}_2$; and $\mathbf{W}_1$ and $\mathbf{W}_2$ stand for the observation noise, assumed Gaussian, zero mean, and white.

Our goal is to estimate the motion parameter vector $\mathbf{p}$. When formulating the *Maximum Likelihood* (ML) estimate of all parameters involved, there is an additional unknown, the scene $\mathbf{S}$, which, however, is easily taken care of. According to the model (1), the ML cost function to minimize is then simply the sum of the overall square error:

$$C_{\mathrm{ML}}(\mathbf{p}, \mathbf{S}) = \iint_{\mathcal{R}} \left\{ \left[\mathbf{I}_1(x,y) - \mathbf{S}(x,y)\right]^2 + \left[\mathbf{I}_2(x,y) - \mathcal{M}(\mathbf{p}^{\#})\mathbf{S}(x,y)\right]^2 \right\} dxdy\,, \tag{2}$$

where the integral is over the support region $\mathcal{R}$. To minimize the ML cost function C_{ML}, given by expression (2), with respect to the unknowns $\mathbf{S}$ and $\mathbf{p}$, we first express the estimate $\widehat{\mathbf{S}}$ of $\mathbf{S}$ in terms of the unknown vector $\mathbf{p}$. Then we insert the estimate $\widehat{\mathbf{S}}$ of the real brightness pattern into expression (2) and minimize with respect to the motion parameter vector $\mathbf{p}$.

By minimizing (2) with respect to the scene brightness pattern $\mathbf{S}$, we get the ML estimate of $\mathbf{S}$ as the average of the brightness pattern of the two images, after registering $\mathbf{I}_2$ according to the image motion (verify, as an exercise):

$$\widehat{\mathbf{S}} = \frac{1}{2}\left[\mathbf{I}_1(x,y) + \mathcal{M}(\mathbf{p})\mathbf{I}_2(x,y)\right]. \tag{3}$$

Replacing $\mathbf{S}$ in the ML cost (2) by $\widehat{\mathbf{S}}$ given by (3), we obtain, after simple algebraic manipulations,

$$C_{\mathrm{ML}}(\mathbf{p}) = \frac{1}{2}\iint_{\mathcal{R}}\left[\mathbf{I}_1(x,y) - \mathcal{M}(\mathbf{p})\mathbf{I}_2(x,y)\right]^2 dx\,dy. \tag{4}$$

Expression (4) makes sense: it shows that the estimate of the motion parameter vector $\mathbf{p}$ is the one that best aligns $\mathbf{I}_2$ with $\mathbf{I}_1$ in the *Least Squares* (LS) sense.

To make explicit the warping of image $\mathbf{I}_2$ according to the motion operator $\mathbf{p}$, we rewrite $\mathcal{M}(\mathbf{p})\mathbf{I}_2(x,y)$ as

$$\begin{aligned}\mathcal{M}(\mathbf{p})\mathbf{I}_2(x,y) &= \mathbf{I}_2\Big(f_x(\mathbf{p};x,y), f_y(\mathbf{p};x,y)\Big)\\ &= \mathbf{I}_2\Big(x + d_x(\mathbf{p};x,y), y + d_y(\mathbf{p};x,y)\Big),\end{aligned} \tag{5}$$

and the ML estimate $\widehat{\mathbf{p}}$ of the motion vector $\mathbf{p}$ minimizing (4) as

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} E(\mathbf{p}), \qquad \text{where} \qquad E(\mathbf{p}) = \iint_{\mathcal{R}} e^2(\mathbf{p};x,y)\,dx\,dy, \tag{6}$$

$$\text{and} \qquad e(\mathbf{p};x,y) = \mathbf{I}_1(x,y) - \mathbf{I}_2\Big(x + d_x(\mathbf{p};x,y), y + d_y(\mathbf{p};x,y)\Big). \tag{7}$$

In expressions (5) and (7), the displacement of the pixel (x,y) between images $\mathbf{I}_1$ and $\mathbf{I}_2$ is denoted by $\mathbf{d}(\mathbf{p};x,y) = [d_x(\mathbf{p};x,y), d_y(\mathbf{p};x,y)]^T$.

To minimize $E(\mathbf{p})$ in expression (6) we use a known technique introduced in reference [2]. This technique uses a *Gauss-Newton method*, where the estimate $\widehat{\mathbf{p}}$ is computed by refining a previous estimate $\mathbf{p}_0$. The initial estimate is usually made the identity mapping, *i.e.*, $\mathcal{M}(\mathbf{p}_0)\mathbf{I}_2 = \mathbf{I}_2 \Leftrightarrow f_x(\mathbf{p}_0;x,y) = x, f_y(\mathbf{p}_0;x,y) = y \Leftrightarrow d_x(\mathbf{p}_0;x,y) = 0, d_y(\mathbf{p}_0;x,y) = 0$.

The error function $e(\mathbf{p}; x, y)$ is approximated by neglecting second and higher order terms of the Taylor series expansion of $e(\mathbf{p}_0 + \boldsymbol{\delta}_p)$,

$$e(\mathbf{p}; x, y) \simeq e(\mathbf{p}_0; x, y) + \boldsymbol{\delta}_p^T \nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y), \tag{8}$$

where $\boldsymbol{\delta}_p = \mathbf{p} - \mathbf{p}_0$ and $\nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y)$ is the gradient of $e(\mathbf{p}; x, y)$ with respect to $\mathbf{p}$ evaluated at $\mathbf{p} = \mathbf{p}_0$.

After inserting the first-order approximation of expression (8) into the cost function (6), the estimate $\widehat{\mathbf{p}}$ is given by

$$\widehat{\mathbf{p}} = \mathbf{p}_0 + \widehat{\boldsymbol{\delta}}_p, \qquad \text{with} \qquad \widehat{\boldsymbol{\delta}}_p = \arg\min_{\boldsymbol{\delta}_p} E\left(\mathbf{p}_0 + \boldsymbol{\delta}_p\right). \tag{9}$$

Equating to $\mathbf{0}$ the gradient of $E\left(\mathbf{p}_0 + \boldsymbol{\delta}_p\right)$ with respect to $\boldsymbol{\delta}_p$, we get the estimate $\widehat{\boldsymbol{\delta}}_p$ as the solution of the linear system

$$\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)\, \widehat{\boldsymbol{\delta}}_p = \boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0), \tag{10}$$

$$\text{where} \qquad \boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0) = \iint_{\mathcal{R}} \nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y) \nabla_{\mathbf{p}} e^T(\mathbf{p}_0; x, y)\, dx\, dy, \tag{11}$$

$$\text{and} \qquad \boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0) = -\iint_{\mathcal{R}} e(\mathbf{p}_0; x, y) \nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y)\, dx\, dy. \tag{12}$$

The vector $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0)$ has the same dimension of $\mathbf{p}$, *i.e.*, $N_p \times 1$ and the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ is square $N_p \times N_p$. Derive (10)-(12), as an exercise.

The error $e(\mathbf{p}_0; x, y)$ and its gradient $\nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y)$ are computed from the spatial and temporal derivatives of the brightness pattern as

$$\begin{aligned} e(\mathbf{p}_0; x, y) &= -i_t(\mathbf{p}_0; x, y), && (13) \\ \nabla_{\mathbf{p}} e(\mathbf{p}_0; x, y) &= -i_x(x, y) \nabla_{\mathbf{p}} d_x(\mathbf{p}_0; x, y) - i_y(x, y) \nabla_{\mathbf{p}} d_y(\mathbf{p}_0; x, y) \\ &= -\nabla_{\mathbf{p}} \mathbf{d}^T(\mathbf{p}_0; x, y) \mathbf{i}_{xy}(x, y), && (14) \end{aligned}$$

where $i_t(\mathbf{p}_0; x, y)$ is the temporal derivative computed by

$$i_t(\mathbf{p}_0; x, y) = \mathbf{I}_2\Big(x + d_x(\mathbf{p}_0; x, y), y + d_y(\mathbf{p}_0; x, y)\Big) - \mathbf{I}_1(x, y), \tag{15}$$

and $i_x(x, y)$ and $i_y(x, y)$ are the spatial derivatives computed from the reference image $\mathbf{I}_1(x, y)$. The 2×1 vector $\mathbf{i}_{xy}(x, y)$ is defined as

$$\mathbf{i}_{xy}(x, y) = \left[\begin{array}{c} i_x(x, y) \\ i_y(x, y) \end{array} \right], \tag{16}$$

and the $N_p \times 2$ matrix $\nabla_{\mathbf{p}} \mathbf{d}^T(\mathbf{p}_0; x, y)$ is defined as

$$\nabla_{\mathbf{p}} \mathbf{d}^T(\mathbf{p}_0; x, y) = \left[\begin{array}{cc} \nabla_{\mathbf{p}} d_x(\mathbf{p}_0; x, y) & \nabla_{\mathbf{p}} d_y(\mathbf{p}_0; x, y) \end{array} \right]. \tag{17}$$

By replacing expressions (13) and (14) into the definitions (11) and (12), we express $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ and $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0)$ in terms of the image derivatives and displacement derivatives as:

$$\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0) = \iint_{\mathcal{R}} \nabla_{\mathbf{p}} \mathbf{d}^T(\mathbf{p}_0) \mathbf{i}_{xy} \mathbf{i}_{xy}^T \nabla_{\mathbf{p}} \mathbf{d}(\mathbf{p}_0)\, dx\, dy, \tag{18}$$

$$\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0) = -\iint_{\mathcal{R}} i_t(\mathbf{p}_0) \nabla_{\mathbf{p}} \mathbf{d}^T(\mathbf{p}_0) \mathbf{i}_{xy}\, dx\, dy, \tag{19}$$

where we omitted the dependence of the integrands on (x, y), for simplicity.

Since the Gauss-Newton method just described approximates the error function $e(\mathbf{p})$ by the truncated Taylor series expansion of $e(\mathbf{p}_0 + \boldsymbol{\delta}_p)$, the initial estimate $\mathbf{p}_0$ must lie in a tight neighborhood of the actual value of the vector $\mathbf{p}$. Since the process is initialized by assuming there is no motion, this means that the motion between consecutive frames must be small, typically sub-pixel motion. Fortunately, a very simple strategy enables to cope with larger displacements: the use of a spatial resolution pyramid. In this pyramid, images $\mathbf{I}_1$ and $\mathbf{I}_2$ are represented at several resolution levels: from the finest one, corresponding to the original resolution, to the coarsest one, where the images will be describe by just a few pixels. Motion is first computed for the coarsest level of resolution (where it can be assumed to be sub-pixel!), and then propagated as initial estimate to the immediately finer level. The process continues until the finer (*i.e.*, full) resolution estimate is obtained.

In order to obtain a reliable convergence of the Gauss-Newton method, the equation system (10) must be well conditioned, *i.e.*, the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$, given by expression (18), must be well conditioned with respect to inversion. A widely used measure for the sensitivity of the solution of the linear system is the *condition number* of the square matrix involved, see reference [4]. The relative error of the solution of a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is approximated by the condition number $k(\mathbf{A})$ of the square matrix $\mathbf{A}$ times the relative errors in $\mathbf{A}$ and $\mathbf{b}$. The condition number depends on the underlying norm used to measure the error. With the common choice of the matrix 2-norm, the condition number of a matrix is given by the quotient of the largest singular value by the smallest singular value, see reference [4]. Since the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ is symmetric and semi-positive definite, their eigenvalues are positive real and coincide with the singular values. The sensitivity of the iterates of the motion estimation algorithm are measured by

$$k(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)) = \frac{\lambda_1(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))}{\lambda_{N_p}(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))}, \tag{20}$$

where the eigenvalues are assumed non-increasingly ordered, *i.e.*, $\lambda_1(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))$ is the largest eigenvalue of $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ and $\lambda_{N_p}(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))$ is its smallest eigenvalue.

If the condition number $k(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))$ is large, the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ is said to be ill-conditioned. In this case, the Gauss-Newton iterates are very sensitive to the noise and the process can not be guaranteed to converge. We will see in sections 4 and 5 what are the practical implications of requiring the condition number $k(\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0))$ to be small. We will discuss there the difficulty of estimating the motion parameters in terms of the variability of image brightness pattern within the support region $\mathcal{R}$.

3 Estimation error

This section studies the statistics of the error in estimating the vector $\mathbf{p}$ of motion parameters. This analysis is local, in the sense that we assume small deviations between the true value of the vector of motion parameters and its estimate. This local analysis is very common in estimation problems. It leads, for example, to the establishment of fundamental bounds like the *Cramér-Rao lower bound* (CRB) for the variance of the estimation error, see reference [5].

In our case, due to the specific structure of the estimator, the small deviation assumption enables the derivation of an expression for the expected noise variance in terms of image spatial gradients. The statistics that we obtain are valid in practice as good approximations to the real statistics if the estimation problem is well conditioned, *i.e.*, if the observations, regardless of the noise level, contain "enough information" to estimate the desired parameters (this imprecise definition can be made precise in terms of the usual signal to noise ratio parameter). This situation is the one in which we are interested because we only use the motion estimates when the corresponding estimation problem is well conditioned in the sense discussed in the previous section.

We denote the actual value of the vector of motion parameters by $\mathbf{p}_a$. The estimate $\widehat{\mathbf{p}}$ is written in terms of a small deviation relative to the actual $\mathbf{p}_a$.

By proceeding in a similar way as done in the previous section, using $\mathbf{p}_a$ instead of $\mathbf{p}_0$ as the central point of the Taylor series expansion, we obtain

$$\widehat{\mathbf{p}} = \mathbf{p}_a + \boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a)\,\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a), \tag{21}$$

where, we recall, the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ and the vector $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a)$ are given by expressions (18) and (19) with $\mathbf{p}_a$ instead of $\mathbf{p}_0$. The random variable $\widehat{\mathbf{p}}$ in expression (21) is a non-linear function of the image derivatives $\{i_t(\mathbf{p}_a), \mathbf{i}_{xy} = [i_x, i_y]^T\}$.

The derivatives i_t and $\mathbf{i}_{xy}$ are random variables – they are noisy versions of the actual values of the scene brightness derivatives. The actual value of $i_t(\mathbf{p}_a)$ is $i_{ta}(\mathbf{p}_a) = 0$ because $\mathbf{p}_a$ is the actual value of the vector of motion parameters. The actual value of $\mathbf{i}_{xy}$ is denoted by $\mathbf{i}_{xya} = [i_{xa}, i_{ya}]^T$. Since the image noise $\mathbf{W}_f(x,y)$ is zero mean, the noise corrupting the derivatives i_t, i_x, and i_y is also zero mean. Furthermore, the noise corrupting the temporal derivative i_t is white because the noise images $\mathbf{W}_1(x,y)$ and $\mathbf{W}_2(x,y)$ are independent. The variance of the noise corrupting the image derivatives is denoted by σ_t^2 for i_t, σ_x^2 for i_x, and σ_y^2 for i_y.

We find the expected value of the estimate $\widehat{\mathbf{p}}$ by computing the mean of expression (21) with respect to the noise of the image derivatives. For small deviations, the first-order approximation of $\mathrm{E}\{\widehat{\mathbf{p}}\}$ is given by the value of expression (21) evaluated at the mean values of the random variables $i_t(\mathbf{p}_a)$ and $\mathbf{i}_{xy}$. Since the mean of $i_t(\mathbf{p}_a)$ is zero, we get $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a) = \mathbf{0}$ and the mean of the estimate $\widehat{\mathbf{p}}$ is

$$\mathrm{E}\left\{\widehat{\mathbf{p}}\right\} = \mathbf{p}_a + \mathrm{E}\left\{\widehat{\boldsymbol{\delta}}_p\right\} = \mathbf{p}_a. \tag{22}$$

Expression (22) states that, to first-order approximation, the estimate $\widehat{\mathbf{p}}$ is unbiased.

The covariance matrix of the estimating error, denoted by $\boldsymbol{\Sigma}_p$, is defined as

$$\boldsymbol{\Sigma}_p = \mathrm{E}\left\{(\widehat{\mathbf{p}} - \mathbf{p}_a)\,(\widehat{\mathbf{p}} - \mathbf{p}_a)^T\right\}. \tag{23}$$

The first-order approximation of the covariance matrix $\boldsymbol{\Sigma}_p$ is related to the partial derivatives of the estimate $\widehat{\mathbf{p}}$ with respect to the random variables involved, *i.e.*, with respect to i_t, i_x, and i_y, and to the variances of those random variables. That result, known from estimation theory, see, *e.g.*, [6], states that the first-order approximation of the covariance matrix of a random vector $\widehat{\mathbf{p}}$ that depends on a set of random variables $\{v_i, i \in V\}$ is given by

$$\boldsymbol{\Sigma}_p = \sum_{k,l \in V} \mathrm{E}\left\{(v_k - \overline{v_k})(v_l - \overline{v_l})\right\} \frac{\partial \widehat{\mathbf{p}}}{\partial v_k} \frac{\partial \widehat{\mathbf{p}}^T}{\partial v_l}, \tag{24}$$

where $\overline{v_i}$ denotes the mean of v_i and the partial derivatives are evaluated at $\{v_i = \overline{v_i}, i \in V\}$.

From expressions (21), (18), and (19), we compute the partial derivatives of $\widehat{\mathbf{p}}$ with respect to the random variables $i_t(x,y)$, $i_x(x,y)$, and $i_y(x,y)$, and evaluate them at the mean values $i_t(x,y) = i_{ta}(x,y) = 0$, $i_x(x,y) = i_{xa}(x,y)$, $i_y(x,y) = i_{ya}(x,y)$. We get

$$\frac{\partial \widehat{\mathbf{p}}}{\partial i_t(x,y)} = -\frac{\partial\left[\boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a)\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a)\right]}{\partial i_t(x,y)} = \boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a)\nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_a))\mathbf{i}_{xya}(x,y), \tag{25}$$

$$\frac{\partial \widehat{\mathbf{p}}}{\partial i_x(x,y)} = -\frac{\partial\left[\boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a)\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a)\right]}{\partial i_x(x,y)} = \mathbf{0}, \tag{26}$$

$$\frac{\partial \widehat{\mathbf{p}}}{\partial i_y(x,y)} = -\frac{\partial\left[\boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a)\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a)\right]}{\partial i_y(x,y)} = \mathbf{0}, \tag{27}$$

where the last two are zero because $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_a) = \mathbf{0}$, since the mean value of i_t is zero.

Since we use a continuous representation of the spatial variables x and y, we write the continuous version of expression (24). Using the fact that the noise corrupting i_t is white and noting that the derivatives in expressions (26) and (27) are zero, we obtain for the covariance $\boldsymbol{\Sigma}_p$,

$$\boldsymbol{\Sigma}_p = \sigma_t^2 \iint_{\mathcal{R}} \frac{\partial \widehat{\mathbf{p}}}{\partial i_t} \frac{\partial \widehat{\mathbf{p}}^T}{\partial i_t}\, dx\, dy. \tag{28}$$

After replacing the derivative of $\widehat{\mathbf{p}}$ with respect to $i_t(x,y)$ given by (25), we get

$$\boldsymbol{\Sigma}_p = \sigma_t^2 \boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a) \iint_{\mathcal{R}} \nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_a)\mathbf{i}_{xya}\mathbf{i}_{xya}^T \nabla_{\mathbf{p}}\mathbf{d}(\mathbf{p}_a)\, dx\, dy \;\; \boldsymbol{\Gamma}_{\mathcal{R}}^{-T}(\mathbf{p}_a). \tag{29}$$

Noting that the integral above is the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}$ evaluated at $\mathbf{p}_a$ (compare to expression (18)), and that the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ is symmetric, we obtain for the error covariance

$$\boldsymbol{\Sigma}_p = \sigma_t^2 \boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a). \tag{30}$$

Expression (30) provides an inexpensive way to compute the reliability of the motion estimates. The matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ is in general unknown because it depends on the actual value $\mathbf{p}_a$ of the unknown vector $\mathbf{p}$. Obviously, the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ can be approximated by the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ used in the iterative estimation algorithm. We note that when the motion model is linear in the motion parameters, as it is the case with the majority of motion models used in practice, the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p})$ becomes independent of the vector $\mathbf{p}$ because the derivatives of the displacement $\mathbf{d}(\mathbf{p})$ involved in expression (18) do not depend on the motion parameters. In this case, the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ does not change along the iterative estimation algorithm. The matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ depends uniquely on the image region $\mathcal{R}$ and $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ will be denoted simply by $\mathbf{\Gamma}_{\mathcal{R}}$. Since the noise variance σ_t^2 is considered to be constant, we measure the error covariance for different regions by comparing the corresponding matrices $\mathbf{\Gamma}_{\mathcal{R}}^{-1}$. For example, the mean square Euclidean distance between the true vector $\mathbf{p}_a$ and the estimated vector $\widehat{\mathbf{p}}$, denoted by σ_p^2, is proportional to the trace of the matrix $\mathbf{\Gamma}_{\mathcal{R}}^{-1}$,

$$\sigma_p^2 = \mathrm{E}\left\{(\widehat{\mathbf{p}} - \mathbf{p}_a)^T (\widehat{\mathbf{p}} - \mathbf{p}_a)\right\} = \sigma_t^2 \mathrm{tr}\left(\mathbf{\Gamma}_{\mathcal{R}}^{-1}\right). \tag{31}$$

These concepts will become clearer in the next two sections where we particularize for the translational motion model and to the affine motion model the results derived in this section and in the previous section.

4 Translational motion

This section studies the translational motion model. The translational motion model is characterized by a constant displacement for all the pixels that fall into the region $\mathcal{R}$. The translational motion model is widely used to estimate the displacement of pointwise features when inferring 3D structure from 2D motion. In this case the region $\mathcal{R}$ is a small square centered in the coordinates of the feature point. The translational model is also frequently used to represent the motion within a larger region $\mathcal{R}$ for special cases of the 3D shape of the scene that is projected onto $\mathcal{R}$ and for special cases of the 3D motion of the camera.

For the translational motion model, the vector $\mathbf{p}$ of motion parameters is defined as

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \tag{32}$$

where p_1 and p_2 determine the displacement $\mathbf{d}(\mathbf{p})$ as

$$\mathbf{d}(\mathbf{p}) = \begin{bmatrix} d_x(\mathbf{p}) \\ d_y(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \mathbf{p}. \tag{33}$$

Motion estimation

The motion parameters are estimated by particularizing to the model of expression (33) the algorithm described in section 2.

To compute the matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ and the vector $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0)$ needed for the Gauss-Newton iterates, we start by making explicit the gradient of the displacement $\mathbf{d}$ with respect to the vector $\mathbf{p}$,

$$\nabla_{\mathbf{p}}\mathbf{d}^T = \begin{bmatrix} \nabla_{\mathbf{p}} d_x & \nabla_{\mathbf{p}} d_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}_{2\times 2}. \tag{34}$$

As advanced at the end of the previous section, the gradient in expression (34) does not depend on the vector $\mathbf{p}$. For this reason, the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ will be independent of $\mathbf{p}_0$ and will remain constant along the iterative process. By replacing expression (34) into expressions (18) and (19), we get

$$\mathbf{\Gamma}_{\mathcal{R}} = \iint_{\mathcal{R}} \mathbf{i}_{xy}\mathbf{i}_{xy}^T \, dx\, dy = \begin{bmatrix} \iint_{\mathcal{R}} i_x^2 \, dx\, dy & \iint_{\mathcal{R}} i_x i_y \, dx\, dy \\ \iint_{\mathcal{R}} i_x i_y \, dx\, dy & \iint_{\mathcal{R}} i_y^2 \, dx\, dy \end{bmatrix}, \tag{35}$$

$$\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0) = -\iint_{\mathcal{R}} i_t(\mathbf{p}_0)\mathbf{i}_{xy} \, dx\, dy = -\begin{bmatrix} \iint_{\mathcal{R}} i_x i_t(\mathbf{p}_0) \, dx\, dy \\ \iint_{\mathcal{R}} i_y i_t(\mathbf{p}_0) \, dx\, dy \end{bmatrix}. \tag{36}$$

Each iteration of the algorithm updates the initial estimate $\mathbf{p}_0$ as $\widehat{\mathbf{p}} = \mathbf{p}_0 + \widehat{\boldsymbol{\delta}_p}$ with $\widehat{\boldsymbol{\delta}_p}$ obtained from expression (10) with $\mathbf{\Gamma}_{\mathcal{R}}$ and $\boldsymbol{\gamma}_{\mathcal{R}}$ given by expressions (35) and (36).

The behavior of the estimation algorithm depends on the condition number of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ of expression (35). The condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ was defined in expression (20) for a general motion model. For the translational motion model, it is the quotient of the first eigenvalue of $\mathbf{\Gamma}_{\mathcal{R}}$ by the second eigenvalue,

$$k(\mathbf{\Gamma}_{\mathcal{R}}) = \frac{\lambda_1(\mathbf{\Gamma}_{\mathcal{R}})}{\lambda_2(\mathbf{\Gamma}_{\mathcal{R}})}, \tag{37}$$

where $\lambda_1(\mathbf{\Gamma}_{\mathcal{R}})$ and $\lambda_2(\mathbf{\Gamma}_{\mathcal{R}})$ are the eigenvalues of $\mathbf{\Gamma}_{\mathcal{R}}$ in expression (35) with $\lambda_1(\mathbf{\Gamma}_{\mathcal{R}}) \geq \lambda_2(\mathbf{\Gamma}_{\mathcal{R}})$. If the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is large, the Gauss-Newton iterates are very sensitive to the noise and the process can not be guaranteed to converge. If the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is small, *i.e.*, if it is close to unity, since $k(\mathbf{\Gamma}_{\mathcal{R}}) \geq 1$, the linear system involved in the Gauss-Newton method is well conditioned.

We discuss when $k(\mathbf{\Gamma}_{\mathcal{R}})$ has large values. To see the influence of the image brightness pattern within region $\mathcal{R}$ on the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ consider that $\iint_{\mathcal{R}} i_x i_y \, dx \, dy = 0$. The matrix $\mathbf{\Gamma}_{\mathcal{R}}$ becomes diagonal and the condition number is simply

$$k(\mathbf{\Gamma}_{\mathcal{R}}) = \frac{\iint_{\mathcal{R}} i_x^2 \, dx \, dy}{\iint_{\mathcal{R}} i_y^2 \, dx \, dy} \tag{38}$$

$$\text{if} \qquad \iint_{\mathcal{R}} i_x^2 \, dx \, dy \geq \iint_{\mathcal{R}} i_y^2 \, dx \, dy \tag{39}$$

or the inverse if the inequality goes in the opposite way. If one of the components of the spatial image gradient is much larger than the other, $k(\mathbf{\Gamma}_{\mathcal{R}})$ becomes large and the equation system (10) is ill-conditioned. The condition

$$k(\mathbf{\Gamma}_{\mathcal{R}}) < \lambda, \tag{40}$$

where λ is a threshold, restricts the brightness pattern within region $\mathcal{R}$ not to have variability along some direction much higher than the variability along the perpendicular direction.

The analysis in the paragraph above explains the well known *aperture problem*. The aperture problem is usually described as the impossibility of estimating locally the 2D motion. In fact, if the region $\mathcal{R}$ contains a single pixel, the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ given by expression (35) is singular; we obtain $\det(\mathbf{\Gamma}_{\mathcal{R}}) = 0$ by removing the integrals from expression (35), and the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is $+\infty$. This happens because the two motion parameters can not be determined by the single constraint imposed by the brightness constancy.

The study of the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ shows that, for particular structures of the brightness pattern, it is very difficult to estimate the 2D motion, even when the region $\mathcal{R}$ contains several pixels.

Expression (38) was obtained with $\iint_{\mathcal{R}} i_x i_y \, dx \, dy = 0$. We should note, however, that the case where

$$\iint_{\mathcal{R}} i_x i_y \, dx \, dy \neq 0 \tag{41}$$

does not correspond to a more general situation. In fact, it can be shown that an appropriate rotation of the brightness pattern makes

$$\int\!\!\int_{\mathcal{R}} i_x i_y \, dx \, dy = 0, \tag{42}$$

without changing the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ – as we would expect, the conditioning of the estimation of the motion of the brightness pattern is independent of 2D rigid transformations of the brightness pattern.

Figure 1 illustrates the dependence of the conditioning of the 2D motion estimation on the structure of the brightness pattern. For each of the eight 10×10 images in Figure 1, we determine the condition number of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$. The condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$, obtained by evaluating expression (37), is on the right side of each image in Figure 1. The texture of the brightness pattern shown on the top left image is such that there is no dominant direction over the entire region $\mathcal{R}$. We expect that the estimation of the 2D motion of a pattern of this kind is very well conditioned. In fact, over the entire image no component of the spatial gradient dominates, and the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ captures this behavior. We get $k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$ – the value of $k(\mathbf{\Gamma}_{\mathcal{R}})$ is close to unity indicating that the linear system involved in the Gauss-Newton iterates of the motion estimation algorithm is well conditioned. In contrast to this case, the texture of the brightness pattern shown in the bottom right image of Figure 1 exhibits a clear dominant direction. It is very hard to perceive the 2D motion of these type of patterns because only the component of the motion that is perpendicular to the dominant direction of the texture is perceived. The condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ captures the indetermination in estimating the 2D motion – it is $k(\mathbf{\Gamma}_{\mathcal{R}}) = 133.82$ indicating that the linear system involved in the Gauss-Newton iterates of the motion estimation algorithm is ill-conditioned. The other images of Figure 1 illustrate intermediate cases.

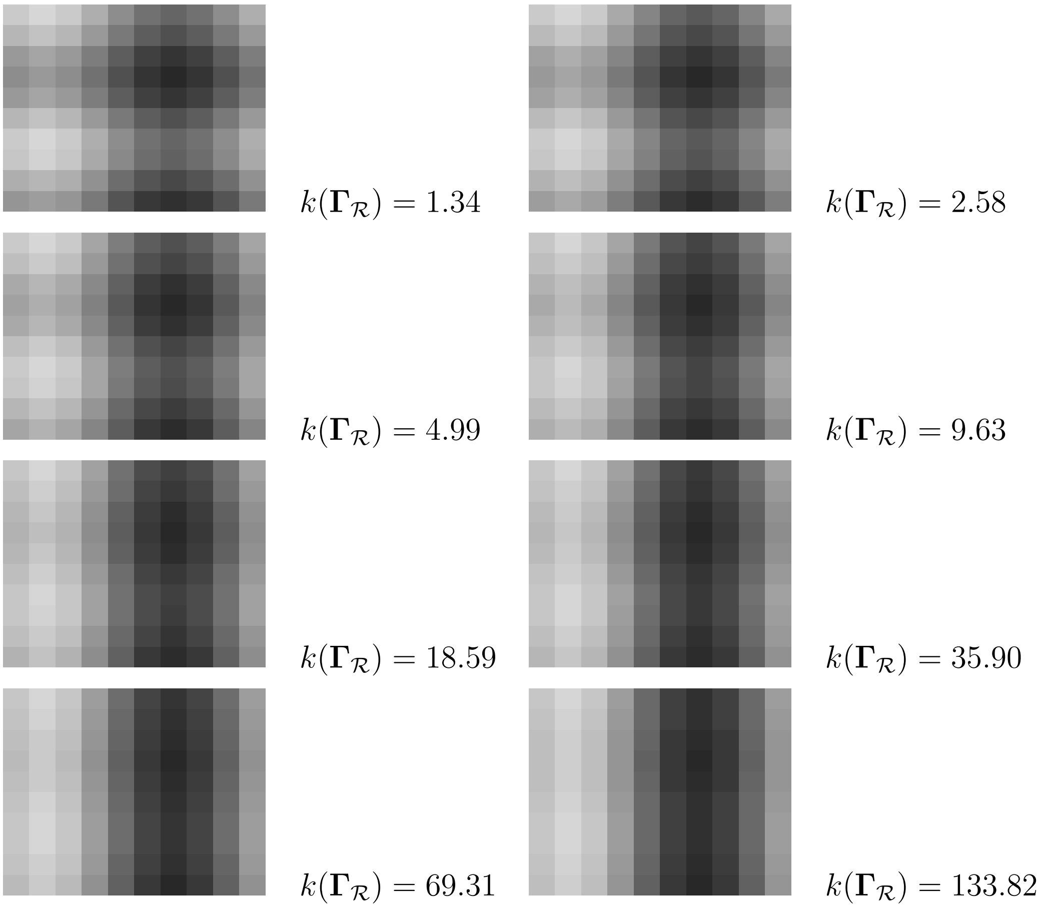
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 2.58$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 4.99$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 9.63$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 18.59$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 35.90$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 69.31$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 133.82$

Figure 1: The dependence of the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ of the matrix involved in the motion estimation algorithm on the structure of the image brightness pattern. When the texture of the brightness pattern exhibits a dominant direction, the motion estimation is ill conditioned – see the bottom right image and the high value of $k(\mathbf{\Gamma}_{\mathcal{R}})$.

Estimation error

The covariance matrix of the estimation error for the translational motion model is given by expression (30) after replacing $\mathbf{\Gamma}_{\mathcal{R}}$ by expression (35),

$$\mathbf{\Sigma}_p = \sigma_t^2 \mathbf{\Gamma}_{\mathcal{R}}^{-1} = \sigma_t^2 \left[\begin{array}{cc} \iint_{\mathcal{R}} i_x^2 \, dx \, dy & \iint_{\mathcal{R}} i_x i_y \, dx \, dy \\ \iint_{\mathcal{R}} i_x i_y \, dx \, dy & \iint_{\mathcal{R}} i_y^2 \, dx \, dy \end{array} \right]^{-1} . \qquad (43)$$

The knowledge of the error covariance matrix $\mathbf{\Sigma}_p$ enables us to compute the reliability of a displacement estimate in an easy way. In fact, in section 3, we saw that the mean square error of the displacement estimate (in the sense of the Euclidean distance) is the trace of the covariance matrix $\mathbf{\Sigma}_p$, see expression (31). In terms of image gradients, we get the following expression for the mean square error, denoted by σ_p^2,

$$\sigma_p^2 = \sigma_t^2 \frac{\int_{\mathcal{R}} i_y^2 \, dx \, dy + \int_{\mathcal{R}} i_x^2 \, dx \, dy}{\int_{\mathcal{R}} i_x^2 \, dx \, dy \, \int_{\mathcal{R}} i_y^2 \, dx \, dy - \left(\int_{\mathcal{R}} i_x i_y \, dx \, dy \right)^2} . \qquad (44)$$

When recovering 3D structure from 2D motion estimates, the estimate of the mean square error σ_p^2 given by expression (44) can be used to weight motion estimates corresponding to different regions [7, 8].

To interpret the mean square error σ_p^2 given by expression (44), let us consider again that the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ is diagonal. This is the general case because, as for the condition number, it can be shown that any non-diagonal matrix $\mathbf{\Gamma}_{\mathcal{R}}$ can be made diagonal without changing σ_p^2, by an appropriate rotation of the image brightness pattern. When $\iint_{\mathcal{R}} i_x i_y \, dx \, dy = 0$, the mean square error σ_p^2 is

$$\sigma_p^2 = \sigma_t^2 \left(\frac{1}{\iint_{\mathcal{R}} i_x^2 \, dx \, dy} + \frac{1}{\iint_{\mathcal{R}} i_y^2 \, dx \, dy} \right) . \qquad (45)$$

Expression (45) states that the error in the estimate of the displacement is proportional to the inverse of the sum of the square components of the image gradient within region $\mathcal{R}$. This coincides with the intuitive notion that the higher the spatial variability of the brightness pattern is, the lower the error in estimating the motion is. As expected, it is also clear that the estimation error decreases when the size of the region $\mathcal{R}$ increases.

Figure 2 illustrates the dependence of the expected square error of the 2D motion estimates on the image brightness pattern. To isolate the estimation error from the eventual ill-posedness of the motion estimation problem, we used brightness patterns that do not have a dominant texture direction, *i.e.*, we used brightness patterns for which the linear system involved in the motion estimation is well conditioned. In particular, we used the brightness pattern of the top left image of Figure 1 to generate all the images of Figure 2 by changing the brightness contrast. The conditioning of the linear system involved in the motion estimation problem does not depend on the brightness contrast, as shown by the constant value of the condition number, $k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$ for all the images in Figure 2.

For each image in Figure 2, we computed the mean square error σ_p^2 by evaluating expression (44). Since the goal is to illustrate the influence of the brightness pattern on σ_p^2, we made $\sigma_t^2 = 1$ when evaluating expression (44). The values obtained for σ_p^2 are shown in Figure 2 on the right side of the corresponding image. The top left image of Figure 2 has a very high brightness contrast. For this reason, we expect that the estimate of the 2D motion of such a pattern is very accurate. In fact, the sum of the square components of the image gradient has a high value and the value of the motion estimation mean square error is low, $\sigma_p^2 = 0.19$. When the brightness contrast decreases, we expect less accurate motion estimates. The values of σ_p^2 in Figure 2 are in agreement with this. The expected square error σ_p^2 increases with the decrease of brightness contrast because the square components of the image gradient decrease. The bottom right image of Figure 2 shows the extreme situation of a pattern with almost zero brightness contrast. For this pattern, the expected mean square estimation error is very high – larger than 60 times the error for the top left image. It is therefore hopeless to try to compute accurate motion estimates for this kind of low contrast patterns. Note that this is due to the fundamental bound on the motion estimation error, not to the conditioning of the linear system involved in the motion estimation algorithm (the condition number $k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$ is close to unity indicating that the linear system is well conditioned).

In summary, for the estimation algorithm to be stable, the two components of the image gradient should not have too radically different magnitude values. With respect to the mean square error of the displacement estimate, we argued that when the magnitude of the components of the image gradient is large, the error is smaller.

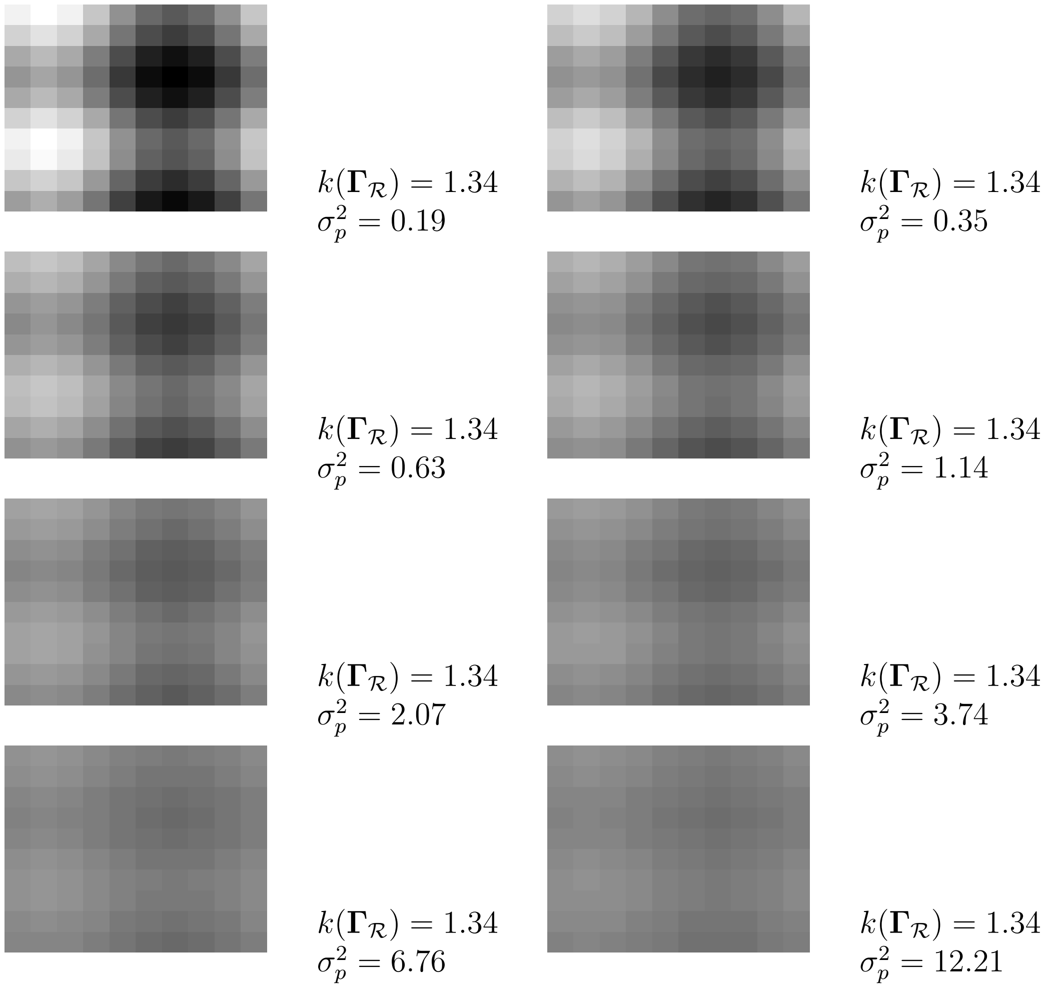
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 0.19$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 0.35$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 0.63$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 1.14$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 2.07$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 3.74$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 6.76$
$k(\mathbf{\Gamma}_{\mathcal{R}}) = 1.34$
$\sigma_p^2 = 12.21$

Figure 2: The dependence of the motion estimation error on the image brightness pattern. The expected square error σ_p^2 increases with the decrease of the brightness contrast.

5 Affine motion

This section studies the affine motion model. The affine model is also widely used in practice. For example, a planar surface moving far away from the camera, undergoing an arbitrary 3D motion, induces an affine motion for the brightness pattern between pairs of images.

The vector $\mathbf{p}$ parameterizing the affine motion model has 6 components,

$$\mathbf{p} = \left[\begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array}\right]^T . \tag{46}$$

The affine displacement $\mathbf{d}(\mathbf{p})$, to which we will also refer to as the affine mapping, is given by

$$\mathbf{d}(\mathbf{p}) = \left[\begin{array}{c} d_x(\mathbf{p}) \\ d_y(\mathbf{p}) \end{array}\right] = \left[\begin{array}{c} p_1 \\ p_2 \end{array}\right] + \left[\begin{array}{cc} p_3 & p_5 \\ p_4 & p_6 \end{array}\right] \left[\begin{array}{c} x - x_c \\ y - y_c \end{array}\right], \tag{47}$$

where x_c and y_c are arbitrary constants that in practice we choose to be the center of the region $\mathcal{R}$ to improve the stability of the estimation algorithm, as will become clear below.

Motion estimation

The affine motion parameters are estimated by using the algorithm described in section 1. To specialize the expressions of matrix $\mathbf{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ and vector $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0)$ involved in the Gauss-Newton iterates to the affine motion model, we start by computing the gradient of the affine displacement with respect to the motion parameters, obtaining

$$\nabla_{\mathbf{p}}\mathbf{d}^T = \left[\begin{array}{cc} \nabla_{\mathbf{p}} d_x & \nabla_{\mathbf{p}} d_y \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ x - x_c & 0 \\ 0 & x - x_c \\ y - y_c & 0 \\ 0 & y - y_c \end{array}\right] . \tag{48}$$

As with the translation motion model, because the affine motion model is linear on the motion parameters, the gradient in expression (48) does not depend on the vector $\mathbf{p}$. The matrix $\mathbf{\Gamma}_{\mathcal{R}}$ will then be independent of $\mathbf{p}_0$ and remain constant along the iterative process. By replacing expression (48) into expressions (18) and (19), we get the following expressions for $\mathbf{\Gamma}_{\mathcal{R}}$ and $\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0)$,

$$\mathbf{\Gamma}_{\mathcal{R}} = \begin{bmatrix} \int_{\mathcal{R}} i_x^2 & \int_{\mathcal{R}} i_x i_y & \int_{\mathcal{R}} \tilde{x} i_x^2 & \int_{\mathcal{R}} \tilde{x} i_x i_y & \int_{\mathcal{R}} \tilde{y} i_x^2 & \int_{\mathcal{R}} \tilde{y} i_x i_y \\ \int_{\mathcal{R}} i_x i_y & \int_{\mathcal{R}} i_y^2 & \int_{\mathcal{R}} \tilde{x} i_x i_y & \int_{\mathcal{R}} \tilde{x} i_y^2 & \int_{\mathcal{R}} \tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{y} i_y^2 \\ \int_{\mathcal{R}} \tilde{x} i_x^2 & \int_{\mathcal{R}} \tilde{x} i_x i_y & \int_{\mathcal{R}} \tilde{x}^2 i_x^2 & \int_{\mathcal{R}} \tilde{x}^2 i_x i_y & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x^2 & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x i_y \\ \int_{\mathcal{R}} \tilde{x} i_x i_y & \int_{\mathcal{R}} \tilde{x} i_y^2 & \int_{\mathcal{R}} \tilde{x}^2 i_x i_y & \int_{\mathcal{R}} \tilde{x}^2 i_y^2 & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_y^2 \\ \int_{\mathcal{R}} \tilde{y} i_x^2 & \int_{\mathcal{R}} \tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x^2 & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{y}^2 i_x^2 & \int_{\mathcal{R}} \tilde{y}^2 i_x i_y \\ \int_{\mathcal{R}} \tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{y} i_y^2 & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_x i_y & \int_{\mathcal{R}} \tilde{x}\tilde{y} i_y^2 & \int_{\mathcal{R}} \tilde{y}^2 i_x i_y & \int_{\mathcal{R}} \tilde{y}^2 i_y^2 \end{bmatrix}, \tag{49}$$

$$\boldsymbol{\gamma}_{\mathcal{R}}(\mathbf{p}_0) = - \begin{bmatrix} \iint_{\mathcal{R}} i_x i_t(\mathbf{p}_0)\, dx\, dy \\ \iint_{\mathcal{R}} i_y i_t(\mathbf{p}_0)\, dx\, dy \\ \iint_{\mathcal{R}} \tilde{x} i_x i_t(\mathbf{p}_0)\, dx\, dy \\ \iint_{\mathcal{R}} \tilde{x} i_y i_t(\mathbf{p}_0)\, dx\, dy \\ \iint_{\mathcal{R}} \tilde{y} i_x i_t(\mathbf{p}_0)\, dx\, dy \\ \iint_{\mathcal{R}} \tilde{y} i_y i_t(\mathbf{p}_0)\, dx\, dy \end{bmatrix}, \tag{50}$$

$$\text{where} \qquad \begin{cases} \tilde{x} = x - x_c \\ \tilde{y} = y - y_c \end{cases}. \tag{51}$$

The stability of the motion estimation algorithm depends on the condition number of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ above. If the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is high, the linear system involved in the motion estimation iterative algorithm is ill conditioned. If the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is low (close to one) that system is well conditioned and the algorithm is stable. To understand the influence of the constants x_c and y_c on the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$, we evaluated $k(\mathbf{\Gamma}_{\mathcal{R}})$ for a simpler case – the one-dimensional (1D) affine motion model. The condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ for the 2D affine model is complex to study because, in opposition to the translational motion model, the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ in expression (49) can not be diagonalized by rotating the image brightness pattern. The condition number of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ involved in the estimation of the parameters of the 1D affine model exhibits the property we want to illustrate for the 2D affine model and leads to a much simpler expression.

The 1D affine motion model is parameterized by the vector $\mathbf{p} = [p_1, p_2]^T$. The 1D displacement is

$$d(\mathbf{p}) = p_1 + p_2(x - x_c). \tag{52}$$

Proceeding as we did for the 2D case, we obtain the 2×2 matrix $\mathbf{\Gamma}_{\mathcal{R}}$ involved in the 1D affine motion estimation,

$$\mathbf{\Gamma}_{\mathcal{R}} = \left[\begin{array}{cc} \int_{\mathcal{R}} i_x^2 \, dx & \int_{\mathcal{R}} (x - x_c) i_x^2 \, dx \\ \int_{\mathcal{R}} (x - x_c) i_x^2 \, dx & \int_{\mathcal{R}} (x - x_c)^2 i_x^2 \, dx \end{array} \right]. \tag{53}$$

The 2×2 semi-positive definite matrix $\mathbf{\Gamma}_{\mathcal{R}}$ in expression (53) is the 1D version of the matrix in expression (49). We obtain the condition number of a generic semi-positive definite 2×2 matrix in terms of the entries of the matrix, by expressing the quotient of the larger by the smaller eigenvalue of the matrix,

$$\mathbf{A} = \left[\begin{array}{cc} a_{11} & a_{12} \\ a_{12} & a_{22} \end{array} \right] \qquad \Longrightarrow \qquad k(\mathbf{A}) = \frac{a_{11} + a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}}{a_{11} + a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}^2}}. \tag{54}$$

In Figure 3 we plot the condition number of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ in expression (53) in terms of the entries of the matrix. We fixed the entry

$$a_{11} = \int_{\mathcal{R}} i_x^2 \, dx = 1. \tag{55}$$

We used expression (54) to evaluate the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ with

$$a_{22} = \int_{\mathcal{R}} (x - x_c)^2 i_x^2 \, dx \tag{56}$$

ranging from 1 to 100, and

$$a_{12} = \int_{\mathcal{R}} (x - x_c) i_x^2 \, dx \tag{57}$$

ranging from -5 to 5. From Figure 3 we can see that the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is small if $\int_{\mathcal{R}} (x - x_c)^2 i_x^2 \, dx$ is close to one and $\int_{\mathcal{R}} (x - x_c) i_x^2 \, dx$ is close to zero. In this case, the linear system involved in the motion estimation algorithm is very well conditioned. If either the value of $\int_{\mathcal{R}} (x - x_c)^2 i_x^2 \, dx$ or the absolute value of $\int_{\mathcal{R}} (x - x_c) i_x^2 \, dx$ are very high, the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ is high and the linear system is ill conditioned. For this reason, the optimal choice for the constant x_c is the center of the region $\mathcal{R}$.

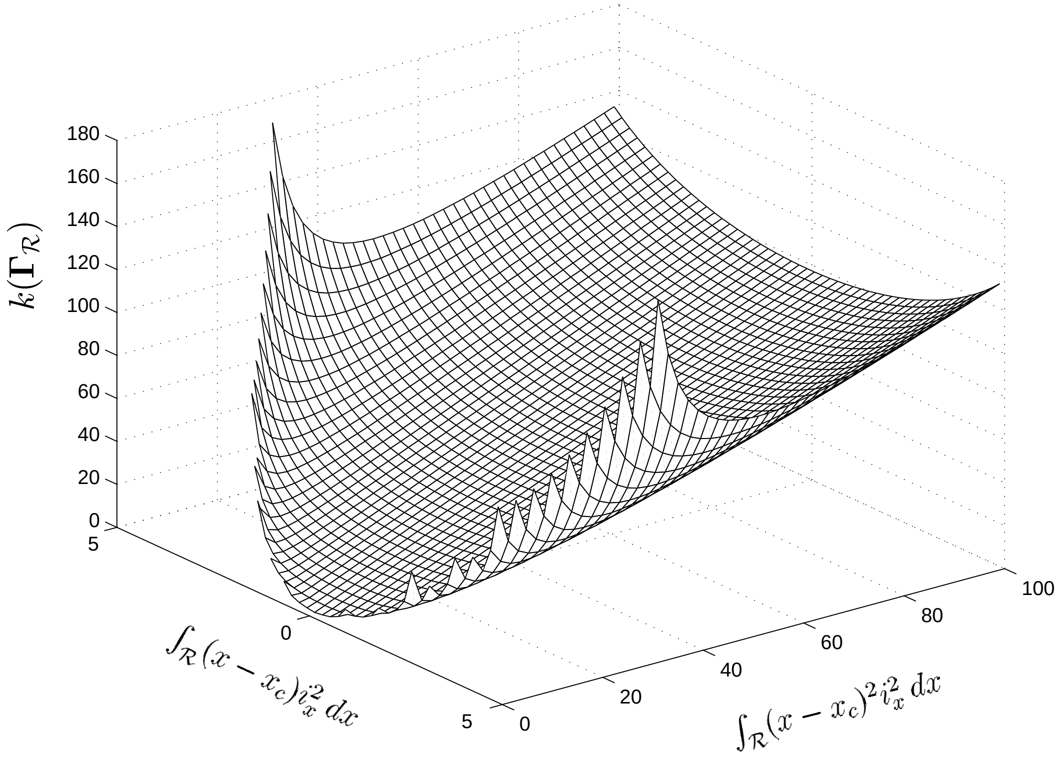

Figure 3: Affine motion estimation. The dependence of the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ on the entries $(1, 2)$ and $(2, 2)$ of the matrix $\mathbf{\Gamma}_{\mathcal{R}}$ of expression (53). The entry $(1, 1)$ was kept fixed, $\int_{\mathcal{R}} i_x^2\,dx = 1$. If the constant x_c is chosen to be far from the center of the region $\mathcal{R}$, the linear system involved in the motion estimation algorithm may be very ill conditioned.

For the 2D affine motion model, the condition number of the 6×6 matrix $\mathbf{\Gamma}_{\mathcal{R}}$ is given by

$$k(\mathbf{\Gamma}_{\mathcal{R}}) = \frac{\lambda_1(\mathbf{\Gamma}_{\mathcal{R}})}{\lambda_6(\mathbf{\Gamma}_{\mathcal{R}})}, \tag{58}$$

where $\lambda_1(\mathbf{\Gamma}_{\mathcal{R}})$ is the largest eigenvalue of $\mathbf{\Gamma}_{\mathcal{R}}$ and $\lambda_6(\mathbf{\Gamma}_{\mathcal{R}})$ is its smallest eigenvalue. We expect the same kind of behavior of the condition number $k(\mathbf{\Gamma}_{\mathcal{R}})$ in terms of the constants x_c and y_c. We then choose the constants x_c and y_c to be the center of the support region $\mathcal{R}$, as noted at the beginning of the section.

Estimation error

The covariance matrix of the estimation error for the affine motion model is given by expression (30) after replacing $\mathbf{\Gamma}_{\mathcal{R}}$ by expression (49),

$$\mathbf{\Sigma}_p = \sigma_t^2 \mathbf{\Gamma}_{\mathcal{R}}^{-1}. \tag{59}$$

The knowledge of the error covariance matrix $\mathbf{\Sigma}_p$ enables us to compute the reliability of a motion parameter estimate in an easy way. References [7] and [8] use the reliability of the estimates of the motion parameters to weight the contribution of those estimates on the recovery of 3D structure. We define the error $\sigma_{\mathcal{P}}^2$ as the mean square error Euclidean distance between the true and estimated values of a subset of the set of motion parameters,

$$\sigma_{\mathcal{P}}^2 = \mathrm{E}\left\{ \sum_{i\in\mathcal{P}} (\widehat{p}_i - p_{ia})^2 \right\}, \tag{60}$$

where $\mathcal{P}$ is the subset of the set of motion parameters and p_{ia} is the actual value of parameter p_i. Since the error $\sigma_{\mathcal{P}}^2$ depends only on the main diagonal entries of the covariance matrix $\mathbf{\Sigma}_p$, we obtain from expression (59),

$$\sigma_{\mathcal{P}}^2 = \sigma_t^2 \sum_{i\in\mathcal{P}} \mathbf{\Pi}_{ii}, \qquad \text{where} \qquad \mathbf{\Pi} = \mathbf{\Gamma}_{\mathcal{R}}^{-1}. \tag{61}$$

In expression (61), the set $\mathcal{P}$ is an arbitrary subset of the parameters of the affine motion model. For example, we can compute the variance of the estimation error of a single parameter p_i by making $\mathcal{P} = \{p_i\}$, or the mean square Euclidean error of the translational component of the affine motion by making $\mathcal{P} = \{p_1, p_2\}$.

6 Summary

In these notes, we studied the estimation of the motion of the brightness pattern between a pair of frames. The motion estimation technique described is global, in the sense that the parameters describing the motion are estimated directly from the all image intensities available, rather than from the displacements of a set of feature points. We discussed the conditioning of the 2D motion estimation problem and derived an expression for the covariance of the estimation error in terms of the image spatial gradients.

We specialized the analysis to two motion models often used in practice – the translational motion model, and the affine motion model. For the translational motion model, we relate the conditioning of the estimation problem to the variability of the spatial brightness pattern. For the affine motion model, we also discuss the influence of the origin of the affine mapping on the conditioning of the estimation problem. For both motion models, we derive expressions for the expected square of the Euclidean distance between the true and estimated values of the parameters.

The algorithm described in these notes is summarized as:

- Build multiresolution pyramids $\left\{\mathbf{I}_1^l, \mathbf{I}_2^l, 1 \leq l \leq L\right\}$ for images$\mathbf{I}_1$ and $\mathbf{I}_2$. The images at resolution $l = 1$ are the original ones, $\mathbf{I}_1^1 = \mathbf{I}_1, \mathbf{I}_2^1 = \mathbf{I}_2$. The coarsest resolution images are $\mathbf{I}_1^L$ and $\mathbf{I}_2^L$.

- Initialize the motion estimate at resolution level L, *i.e.*, $\mathbf{p}_0^L$, as the identity mapping.

- For $l = L$ down to 1 do
  - Repeat
    - Update estimate of motion parameters at level resolution level l, *i.e.*, $\mathbf{p}^l$, using the solution of (10), with $\mathbf{\Gamma}_{\mathcal{R}}$ and $\boldsymbol{\gamma}_{\mathcal{R}}$ given by (18,19) (expressions (35,36) or (49,50) in the case of the translational or affine motion model).
  - Until convergence.
  - Convert the current motion estimate $\mathbf{p}^l$ to the next resolution level $\mathbf{p}_0^{l-1}$ (just take into account the change of scale).

Algorithms that extend in several ways the ones presented here have also been proposed, *e.g.*, using more general motion models [9], simultaneous registration of multiple images [10], or dealing with differently exposed photographs [11].

As a final illustration, we used affine-motion version of the algorithm described in these notes (actually, its extension described in [11]) to align the images of Figure 4.

We represent in Figure 5, from left to right, top to bottom, the evolution of the registration of the right image according to successive estimates of the registration parameters in $\mathbf{p}$ (and exposure correction parameters). The top

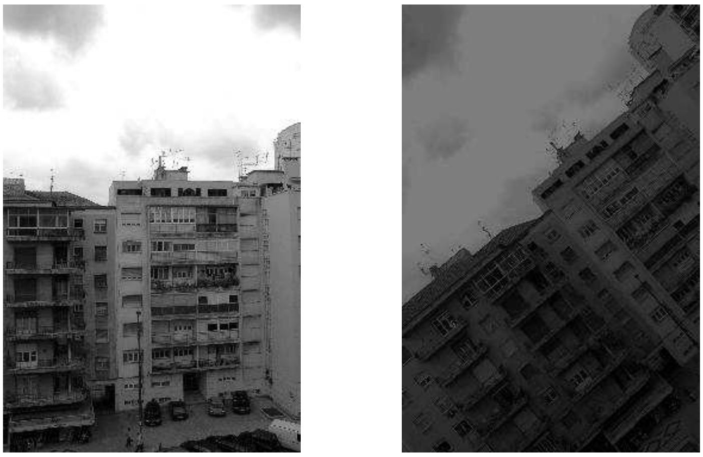

Figure 4: Two photographs of the same scene. Left: “natural” orientation and exposure. Right: tilted and mush darker view.

left image of of Figure 5 shows the lower resolution version of the right image of Figure 4, see how the orientation and exposure of these images is the same. As the registration process evolves, the orientation (and the exposure) of the successive images in Figure 5 converges to the ones that best match the left image of Figure 4. The final result, obtained at the original full resolution, is shown in the bottom right image of Figure 5.

In Figure 6, we represent the composition of the two images of Figure 4, after simultaneous registration (and exposure correction) *i.e.*, the mosaic obtained by merging the left image of Figure 4 with the bottom right image of Figure 5.

References

[1] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17, 1981.

[2] James R. Bergen, Patrick Anandan, Keith J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Proc. of the European Conf. on Computer Vision*, Springer-Verlag Lectures Notes in Computer Science, Santa Margherita Ligure, Italy, 1992.

[3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Prentice Hall, 1992.

[4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins Series in Mathematical Sciences. The Johns Hopkins University Press, second edition, 1989.

[5] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I: Detection, Estimation, and Linear Modulation Theory.* John Wiley & Sons, 1968.

[6] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* Electrical & Electronic Engineering Series. McGraw Hill, third edition, 1991.

[7] P. Anandan and Michal Irani. Factorization with uncertainty. *Int. Journal of Computer Vision*, 49(2-3), 2002.

[8] Pedro M. Q. Aguiar and José M. F. Moura. Rank 1 weighted factorization for 3D structure recovery: algorithms and performance analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[9] Steve Mann and Rosalind W. Picard. Video orbits of the projective group: a simple approach to featureless estimation of parameters. *IEEE Trans. on Image Processing*, 1997.

[10] Bernardo E. Pires and Pedro M. Q. Aguiar. Featureless global alignment of multiple images. In *Proc. of IEEE Int. Conf. on Image Processing*, Genova, Italy, 2005.

[11] Pedro M. Q. Aguiar. Unsupervised simultaneous registration and exposure correction. In *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta GA, USA, 2006.

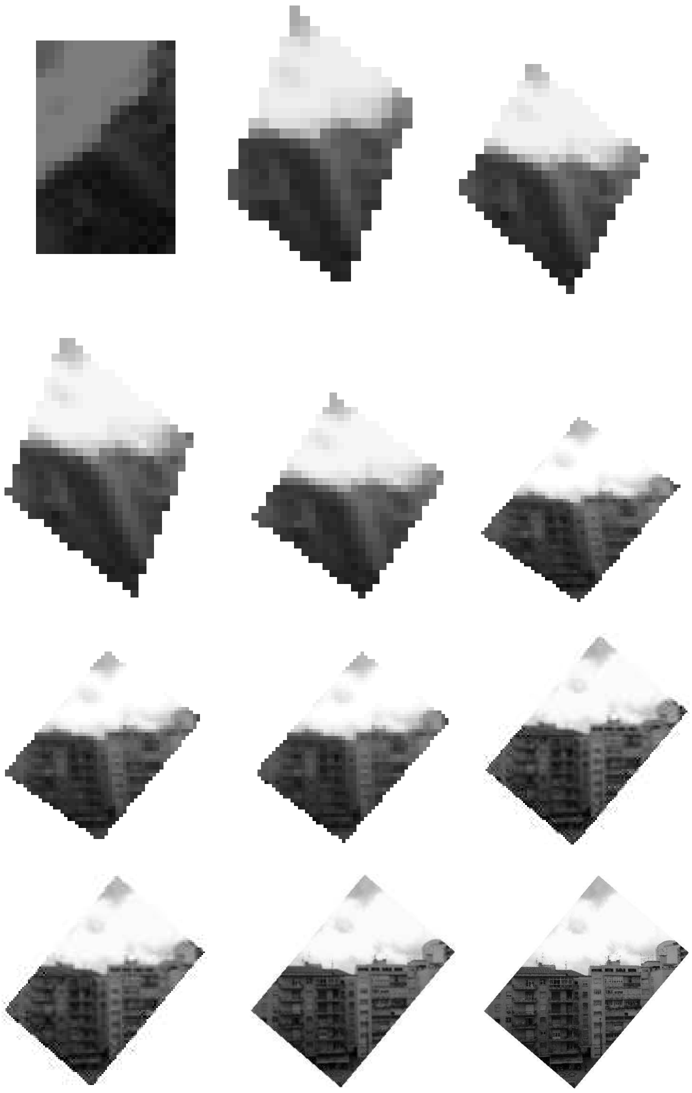

Figure 5: Evolution of the motion estimation algorithm when processing the pair of images of Figure 4. From left to right, top to bottom, we represent the registration of the right image, at increasing resolution levels, according to the estimates of the parameters obtained as the algorithm evolves.

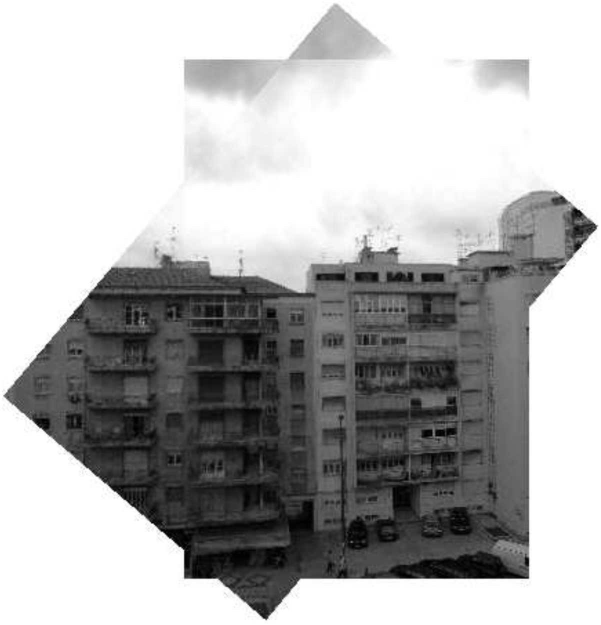

Figure 6: Composition of the images in Figure 4, using our method for simultaneous registration and exposure correction.