This is an author produced version of a paper presented at SSAB’02, Swedish symposium on Image Analysis, 2002 års Svenska Symposium i Bildanalys, 7-8 mars, LTH, Lund

Citation for the published paper:

Niclas Börlin

Adaptive least squares matching as a non-linear least squares optimization problem
Abstract

Adaptive Least Squares Matching (ALSM) is a powerful technique for precisely locating objects in digital images. The method was introduced to the photogrammetric community by Gruen in 1985 and has since been developed further. The purpose of this paper is to study the basic ALSM formulation from a least squares optimization point of view. It turns out that it is possible to describe the basic algorithm as a variation of the Gauss-Newton method for solving weighted non-linear least squares optimization problems. This opens the possibility of applying optimization theory on the ALSM problem. The line-search algorithm for obtaining global convergence is especially described and illustrated.

Key words

Least squares matching, Gauss-Newton algorithm, Photogrammetry, Line search algorithms.

1 Introduction

A central problem in photogrammetry is that given multiple images of an object, locate the 2-d positions of the same 3-d feature in multiple images. These position may then be used to measure the object position and/or to calibrate the camera system.

Adaptive Least Squares Matching (ALSM) was introduced by Gruen in 1985[5] as a method for addressing this problem. ALSM uses a patch, or template, which is adjusted to match a part of an image. The template may be synthetic or taken from a real image. The adjustments of the template may be split in two parts, radiometrical and geometrical. The radiometrical adjustments are introduced to handle differences in brightness and/or contrast between the template and the image. The geometrical adjustments are introduced to handle the differences in the projected geometry of a given feature. In the basic ALSM formulation, an affine transformation of the template geometry is used, which is a good approximation if the object-camera distance is large compared to the template size, which is often the case.

Several variations of the basic method exist: The geometrical transformation may be restricted to shift-only, rotate-only, or shift-and-rotate (no shear). Application knowledge may be used to introduce geometrical position constraints, digital surface model constraints, image feature constraints, etc. For a description of variations, see e.g. [1, 4, 6].

2 Adaptive Least Squares Matching

Photogrammetric formulation

In photogrammetric terms, the ALSM problem is described using a statistical estimation model and is formulated as follows (cf. [4], equations (8.2.1)–(8.2.13)):

\[ f(x, y) - e(x, y) = g(x, y), \]

where \( f(x, y) \) is the template, \( e(x, y) \) is noise, and \( g(x, y) \) is the image data in the points \( (x, y) \). The function \( g(x, y) \) will be called the “picture” and is not directly available, but is formulated as a re-sampling of a “real” image \( g^0(x, y) \). The template may in theory be infinite but is typically small compared to the picture. Under affine transformation, the relation between the coordinates \( x, y \) in the picture and the coordinates \( x_0, y_0 \) in the real image is described by

\[ \begin{align*}
    x &= a_{11} + a_{12} x_0 + a_{21} y_0, \\
    y &= b_{11} + b_{12} x_0 + b_{21} y_0.
\end{align*} \]

Since the system (1) is non-linear, it is linearized into

\[ f(x, y) - e(x, y) = g^0(x, y) + \frac{\partial g^0(x, y)}{\partial x} dx + \frac{\partial g^0(x, y)}{\partial y} dy, \]

where

\[ dx = \frac{\partial x}{\partial p_i} dp_i, \quad dy = \frac{\partial y}{\partial p_i} dp_i, \]
and \( p_i \) is the \( i \)th transformation parameter in Equation (2), i.e.
\[
\begin{align*}
\frac{dx}{d} &= da_{11} + x_0 da_{12} + y_0 a_{21}, \\
\frac{dy}{d} &= db_{11} + x_0 db_{12} + y_0 b_{21}.
\end{align*}
\]

Using the simplified notations
\[
g_x = \frac{\partial g^0(x, y)}{\partial x}, \quad g_y = \frac{\partial g^0(x, y)}{\partial y}
\]
and adding two radiometric parameters \( r_s \) (shift) and \( r_t \) (scale) to Equation (3) yields the complete system equations
\[
f(x, y) - e(x, y) = g^0(x, y) + g_x x_0 da_{11} + g_x x_0 da_{12} \\
+ g_y y_0 da_{21} + g_y db_{11} + g_y x_0 db_{12} \\
+ g_y y_0 db_{21} + r_s + g^0(x, y)r_t.
\]

Combining all parameters of Equation (5) into the parameter vector
\[
x = [da_{11}, da_{12}, da_{21}, db_{11}, db_{12}, db_{21}, r_s, r_t]^T,
\]
their coefficients in the design matrix \( A \), and the vector difference \( f(x, y) - g^0(x, y) \) in \( \ell \), the observation equations are obtained in classical (photogrammetric) notation as
\[
\ell - e = Ax.
\]

Together with the statistical expectation operator \( E \) and the assumptions
\[
E(e) = 0, \quad E(ee^T) = \sigma^2 P^{-1},
\]
Equation (6) is called a Gauss-Markov estimation model.

The least squares estimation of system (6) is
\[
\hat{x} = (A^T PA)^{-1} A^T P \ell.
\]

Since the original problem (1) is non-linear, the final result is obtained iteratively. The starting approximations used are
\[
x^0 = [0, 1, 0, 0, 0, 1, 0, 0]^T,
\]
 corresponding to the following first set of coordinates
\[
x_i = x_{0i}, \quad y_i = y_{0i}, \quad i = 1, \ldots, m,
\]
where \( m \) is the number of pixels in the template. Note that with the starting approximation (8), the vector \( \ell = f(x, y) - g^0(x, y) \) may be interpreted as the starting residual.

After the solution vector (7) has been obtained, the transformation (2) is applied and \( g(x, y) \) is recalculated as a new re-sampling of \( g^0(x, y) \) over the new set of coordinates \((x_i, y_i)\). This may be interpreted as adding the “update vector” \( \hat{x} \) to the current approximation as
\[
x^{k+1} = x^k + \hat{x}^k,
\]
where \( x^k \) is the parameter approximation vector at iteration \( k \) and \( \hat{x}^k \) is the solution vector (7) with the design matrix \( A \) and the difference vector \( \ell \) recalculated at each step. The iteration is terminated when the update vector elements fall below a certain size.

## 3 Non-linear optimization

This section gives a brief description of some properties of non-linear problems and methods. A more elaborate discussion is found in [2]. The notation and terminology used is from non-linear optimization and numerical linear algebra, but references to the ALSM formulation are made. For more background on non-linear optimization, see e.g. [3, 7].

Unless stated otherwise, in the following, the variable \( x \) represents the parameters to be estimated whenever it is written alone. When written as \((x, y)\), it represents an \( x \) coordinate.

An important subclass of unconstrained optimization problems is the least squares optimization problem
\[
\min_x f(x) = \min_x \frac{1}{2} \sum_{i=1}^m f_i(x)^2 = \min_x \frac{1}{2} \| F(x) \|^2
\]
\[
= \min_x \frac{1}{2} F(x)^T F(x),
\]
where the residual function \( F(x) \in \mathbb{R}^m \) is a vector-valued function and \( m \) is the number of observations. The residual function is assumed to be everywhere twice continuously differentiable, and is often formulated as “model minus data”, i.e.
\[
F(x) = G(x) - d,
\]
where the function \( G \) is a model of what we are observing and \( d \) is vector of observations. In the ALSM case, the vector \( d \) correspond to the template \( f(x, y) \), the function \( G(x) \) correspond to the picture \( g(x, y) \), and \( F(x) \) may be interpreted and the residual vector \( \ell \) (with opposite sign). Furthermore, with the starting approximation vector \( x_0 \) defined as in Equation (8), \( G(x_0) \) is equivalent to \( g^0(x, y) \).

The corresponding weighted least squares optimization problem is formulated as
\[
\min_x f(x) = \min_x \frac{1}{2} \| W F(x) \|^2 = \min_x \frac{1}{2} (F(x)^T W F(x)),
\]
where \( W \) is a symmetric positive definite weight matrix, and \( L^T L = W \). The weight matrix \( W \) correspond to the \( P \) matrix in Equation (6).

The gradient \( \nabla f(x) \) of a weighted non-linear least squares function is
\[
\nabla f(x) = J(x)^T W F(x),
\]
where the Jacobian \( J(x) \) of \( F \) at \( x \) contain all partial derivatives such that
\[
J(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j}.
\]
The Jacobian \( J(x) \) correspond to the design matrix \( A \) in Problem (6).
The Gauss-Newton method

The classical method for solving Problem (12) is the Gauss-Newton method, where the iteration sequence

\[ x_{k+1} = x_k + p_k \]  

(14)
is constructed and \( p_k \) is the solution of

\[ J_k^T W J_k p_k = -J_k^T W F_k, \]

(15)with \( J_k = J(x_k) \) and \( F_k = F(x_k) \). The solution of Equation (15) is

\[ p_k = (J_k^T W J_k)^{-1} J_k^T W (-F_k). \]

(16)

By simple substitution

\[ \hat{x} = p_k, \quad A = J_k, \quad P = W, \quad \ell = -F_k, \]

(17)we find that if we express the ALSM estimation model (1) as Equation (11), the ALSM problem is equivalent to a weighted least squares optimization problem (12). Furthermore, the ALSM update (7) is equivalent to the Gauss-Newton update (16). Thus, iterating the calculations described in Section 2 is equivalent to solving Problem (12) with the Gauss-Newton method.

Obtaining convergence

It is known that the Gauss-Newton method is not always locally convergent, i.e. under some circumstances it cannot be guaranteed to converge toward a solution, no matter how close to the solution we start.

However, it is possible to apply a simple modification to the Gauss-Newton method to obtain a method which for most cases is also globally convergent, i.e. it will converge toward a local minimum from any starting point.

Consider the modified iteration sequence

\[ x_{k+1} = x_k + \alpha_k p_k, \]

(18)where \( p_k \) is calculated as before and \( \alpha_k \) is chosen by the following line search algorithm: Select \( \alpha_k > 0 \) as the first value of the sequence \( 1, \frac{1}{2}, \frac{1}{4}, \ldots \) such that the new point \( x_{k+1} \) satisfies the Armijo condition

\[ f(x_k + \alpha_k p_k) \leq f(x_k) + \mu \alpha_k \nabla f(x_k)^T p_k, \]

(19)for a given constant \( \mu, 0 < \mu < 1 \). This algorithm is called the damped Gauss-Newton method. In this context, \( p_k \) is called a search direction and \( \alpha_k \) is called a step length.

The interpretation of the damped Gauss-Newton method is that we search along the search direction \( p_k \) to find out how large step \( \alpha_k \) along \( p_k \) we should take in order to get to a better point \( x_{k+1} \) than \( x_k \), as measured by the objective function \( f(x) = 1/2 F(x)^T F(x) \). Far from the solution, the line search algorithm works as a “safe guard”, making sure that we do not take a long step without getting a corresponding reduction in the size of the residual. Furthermore, if the Jacobian is ill conditioned near the solution, it is not uncommon that by blindly taking full steps \( \alpha_k = 1 \), we will get successive search directions \( p_k \) and \( p_{k+1} \) that are almost opposite and we get oscillations in one or more of the parameters. With a line search algorithm, such oscillations may be “damped out”. In both cases, the line search improves the robustness of the algorithm.

The conventional Gauss-Newton method may be seen as a special — undamped — case of the damped method, where we always take a full step length \( \alpha_k = 1 \), irrespective if we get a smaller residual or not.

The damped Gauss-Newton method is not the only globally convergent method for a weighted non-linear least squares problem, but it has the advantage of being close to the original ALSM algorithm and being easy to implement in an efficient way. For other possible modifications, see e.g. [3, 7].

Interpolation issues

The line search improves the robustness of the ALSM algorithm irrespectively of how the picture \( g(x, y) \) is interpolated. However, if the image is interpolated bilinearly, the picture model \( G(x) \) and hence the objective function \( f(x) \) is no longer everywhere twice continuously differentiable. As illustrated in Figure 1, the pixel value \( g(x, y) \) at the image coordinates \( (x, y) \) is approximated by a weighted sum of the surrounding pixel values. Inside each patch, the function \( g(x, y) \) is twice continuously differentiable, but at the patch borders, the partial derivatives may be discontinuous. Hence, the theoretical foundation for the Gauss-Newton algorithm is no longer valid. Note that this is especially the case for the starting point \( x_0 \). In practice, the calculated search direction will be dependent on how this discontinuity is handled in the implementation. In most cases we can still expect convergence from such a point, but this only adds to the importance of a “safe guarding” algorithm such as the line search.

Figure 1: Bilinear interpolation.
4 An example

The difference between an undamped and a damped Gauss-Newton algorithm is illustrated in the following example. The template in Figure 2 is matched to the image in Figure 3. The white block in the template is 11×11 pixels with a 4 pixel border. The white blocks in the image are approximately 19 pixels wide with 12 pixel separation. The template is binary but the image has been low-pass filtered to smooth the edges and create a more difficult matching problem.

The first iteration is illustrated in Figure 3 (left). The solid line is the starting position \( x_0 \). The dashed line indicates the next position \( x_1 = x_0 + p_0 \) of the patch. In this case the line search algorithm would accept a full step \( \alpha_0 = 1 \) since the residual length is reduce by about 50%.

In the second iteration, illustrated in Figure 3 (right), the current position \( x_1 \) is indicated with the solid line. The dotted line indicates the new position \( x_2 = x_1 + p_1 \) that the undamped Gauss-Newton would accept. However, since the residual is actually increased by 50% at \( x_2 \), the line search algorithm would not accept a full step \( \alpha_1 = 1 \). Instead a step length \( \alpha_1 = \frac{1}{2} \) would be accepted, since it reduces the size of the residual by about 40%. The position of the “damped” patch at \( x'_2 = x_1 + \frac{1}{2}p_1 \) is indicated with dashed lines.

The undamped algorithm would take the dotted \( x_2 \) as the new position of the patch. After another 48 iterations it would still be oscillating about the position shown in Figure 4 (left) with the largest element \( da_{11} \) alternating between \( \approx \pm 0.24 \).

In contrast, the damped algorithm would continue from \( x'_2 \) and converge toward the solution in Figure 4 (right) in only 7 more iterations. In this case, full steps are acceptable in each of the subsequent step. Thus, the extra cost of obtaining convergence was only one extra function evaluation at iteration 2.

5 Summary and future work

This example illustrates the usefulness of a line search strategy in solving an Adaptive Least Squares Matching problem. In this case, the undamped algorithm converged toward an incorrect solution. However, in cases of images with repetitive patterns, it is also believed that the likelihood of converging toward the minimum nearest to the starting point is increased. This will be investigated in the future.

Furthermore, this paper illustrates but one example of the large potential of combining the theories and experience of two mature research fields such as photogrammetry and non-linear least squares optimization. This potential will also be explored further in the future.

References