Image Warping for 3–D Reconstruction
Robustness and Efficiency

Joachim Hornegger[1,2] and Carlo Tomasi[2]

[1] Siemens Medical Solutions, AXE 1,
Siemensstr. 1, 91301 Forchheim, Germany
[2] Robotics Laboratory, Department of Computer Science,
Stanford, CA 94305, USA
Email: {jh,tomasi}@robotics.stanford.edu

Abstract. Many problems in medical imaging are stated in terms of parameter estimation. Given a proper parametric function local or global optimization procedures are applied to compute the best fit between the chosen model and observed measurements. The mathematical formalization of the tackled problem decides on the success of the final algorithm. We introduce a framework for practitioners which is useful for the definition of parameter estimation problems. The discussion of these techniques is guided by a concrete application: all the theory is motivated and studied considering the image warping task required for 3-D reconstruction based on linear projective mappings.

1 Introduction

A major goal of computer vision is to infer 3-D structure from image data. Almost all reconstruction algorithms are stated as multiple view problems and rely on camera models which are linearly projective [1]. In most cases the optical acquisition device causes a nonlinear image distortion which is not sufficiently modeled by a linear mapping in the projective space. CCD cameras, for instance, cause radial and tangential distortions which are caused by improper assembly and by the manufacturing process of lenses [1]. In X-ray systems image distortion is basically caused by the earth magnetic field. The implied distortion function is neither linear nor radial symmetric. In the presence of image distortion it is important that the implementation of reconstruction algorithms provides a reliable distortion correction module such that straight lines in 3-D are still mapped to straight lines in the projection image.

Especially in medical applications there is a huge demand on the reliability of algorithms. It is not sufficient to provide any solution, there is an obvious need to implement the best possible algorithm with guaranteed success and predictable runtime behavior. For that reason we will study the problem of image warping in terms of robustness and efficency. This contribution provides important and rather universal tools for the design and implementation of algorithms which fit the high needs of medical applications. It is observed in the literature that many ill-conditioned problems are insufficiently solved because the applied parametric model and the input data are not carefully designed [1].

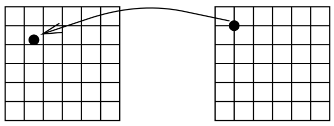

Fig. 1. Image warping: original image $\boldsymbol{f}$ (left), warped image $\boldsymbol{f}'$ (right)

2 Principle of Image Warping

As illustrated in Fig. 1 the process of image warping maps an image to another image. For each pixel (x', y') of the new image $\boldsymbol{f}'$ we have to compute the corresponding point (x, y) of the original (distorted) image $\boldsymbol{f}$. In a second step the intensity value of $\boldsymbol{f}'$ at (x', y') has to be determined. Due to the fact that the corresponding x and y are generally no grid points the required intensity has to be computed by interpolating gray-levels of the local neighborhood. The discussion of interpolation is omitted here. For further details we recommend the survey paper [2]. In image processing the warp function is not defined by users but given due to the physics of the acquisition device. The computation of the warp function from observations is called calibration. Another interesting problem is the selection of the direction of the above mapping: do we map pixels from $\boldsymbol{f}'$ to $\boldsymbol{f}$ and run interpolation in $\boldsymbol{f}$ or vice versa? Discrete mathematics tells us that you better sample the output function $\boldsymbol{f}'$.

The design of an image warping algorithm for distortion correction requires the solution of basically three sub-problems: 1. *Model design:* definition of a proper (parametric) warp function. 2. *Calibration:* estimation of model parameters from observations; this also includes the optimal design of a calibration pattern 3. *Usage:* application of the computed model.

3 Parametric Modeling of Warp Functions

The mapping between image coordinates of $\boldsymbol{f}'$ and $\boldsymbol{f}$ can be defined locally or globally. We can either decompose the image into blocks and map these blocks separately or warp the complete image by a single global function for each coordinate. In both cases mappings can be approximated, for instance, by bivariate polynomials of total degree d:

$$x = X(x', y') = \sum_{\substack{i,j \\ i+j \leq d}} u_{i,j} b_i(x') b_j(y') = \sum_{i=0}^{d} \left(\sum_{j=0}^{d-i} u_{i,j} b_j(y') \right) b_i(x') \ . \qquad (1)$$

Here $b_k(x'), k \geq 0$ span the considered polynomial space. The function $Y(x', y')$ to computed corresponding y-coordinates is defined analogous; its coefficients are denoted by $v_{i,j} \in \mathbb{R}$. An obvious but less recommended base is $b_k(x') = x'^k$.

Stability. The best choice for $b_k(x')$ are orthogonal polynomials where the coefficients are known to be mutually independent. Uncertainties in single coefficients do not affect other parameters. In terms of statistics this parameterization is called *stable.* The covariance matrix of the estimated parameter vector is a diagonal matrix. Thus the covariance matrix serves as a witness for the stability of a certain parameterization.

Fairness. Besides stability, fairness of the parameterization is another important issue in defining regression models. The sensitivity of the algorithm's output should not depend on the chosen coordinate system. Therefore we call a parameterization *fair* if any rigid transform of the space implies an orthogonal transform of parameters [3]. The above introduced polynomial (1) is an example for an unfair parameterization. The orientation of the coordinate system affects the numerical sensitivity of the considered problem. Thus high variances in estimates are not necessarily an intrinsic property of the problem.

Normalization. The concept of fairness does not cover the problem of scaling. It is expected that scaling of data does not affect the final result of the parameter estimation algorithm. From a theoretical point of view this assumption holds but in practice we have to deal with image noise and limited precision arithmetics. Therefore *normalization* of measurements is an important issue. Scaling of input data often affects the conditioning of the problem to be solved and cannot be neglected [1].

4 Calibration

The estimation of the coefficient vectors $\boldsymbol{u} = (u_{i,j})_{i+j\leq d}$ and $\boldsymbol{v} = (v_{i,j})_{i+j\leq d}$ of above introduced warp functions requires calibration.

4.1 Maximum Likelihood Estimation

We introduce a calibration pattern where the world coordinates of points are precisely known. We denote these points by $(x'_n, y'_n), n = 1, 2, \ldots, N$. Image measurements are the warped points of the calibration pattern, i.e. $(x_n, y_n), n = 1, 2, \ldots, N$. The observed feature points and the knowledge of ideal calibration points allow the estimation of parameters in (1). We just take those parameters which minimize the deviation between measurements and primitives of the calibration pattern. The estimates crucially depend on the choice of the cost function. Without prior knowledge the best estimates of $\boldsymbol{u}$ and $\boldsymbol{v}$ result from a maximum likelihood estimate [3]. Let us assume that the segmented feature points (x_n, y_n) are isotropic and Gaussian random measures where variances are equal to 1. If $(x'_n, y'_n), n = 1, 2, \ldots, N$ denote the corresponding (ideal) coordinates of the calibration pattern the unknown parameter vector $\boldsymbol{u}$ results from the least square estimate:

$$\widehat{\boldsymbol{u}} = \mathrm{argmin}_{\boldsymbol{u}} \sum_{n=1}^{N} (X(x'_n, y'_n) - x_n)^2 \ . \tag{2}$$

In this case the likelihood function and the associated optimization task define a linear mapping from the parameter vector $\boldsymbol{u}$ to measurement vector $\boldsymbol{m}$ and vice versa, i.e. $\boldsymbol{m} = \boldsymbol{A}\boldsymbol{u}$. The components of the matrix $\boldsymbol{A}$ are nonlinear functions of ideal points of the calibration pattern. If noisy measurements are considered as random variables, the optimization problem defines a transform $\mathcal{T}$ of random variables. The fairness of the chosen parameterization can be checked by considering the Jacobian $\boldsymbol{J}_{\boldsymbol{u}}$ of the transform $\mathcal{T}$. In case of a fair parameterization the singular values of matrix $\boldsymbol{J}_{\boldsymbol{u}}$ will remain constant by rigid changes of coordinates [3]. Any residual function implies a mapping $\mathcal{T}$ which can be linearly approximated at its minimum using the implicit function theorem. Therefore there is no need to have a closed form $\mathcal{T}$ to compute the Jacobian $\boldsymbol{J}_{\boldsymbol{u}}$.

4.2 Linear Estimators and Normalization

The mapping introduced in (1) is linear in the coefficients of the polynomial. As mentioned above the least square estimate can be solved in closed form. The parameter vector is computed by: $\widehat{\boldsymbol{u}} = \boldsymbol{A}^{\dagger}\boldsymbol{m}$, where $\boldsymbol{A}^{\dagger}$ denotes the pseudo inverse of $\boldsymbol{A}$. Obviously the matrix $\boldsymbol{A}$ and its pseudo inverse depend on measurements only.

The condition number $\kappa(\boldsymbol{A}^T\boldsymbol{A})$ of the non-singular matrix $\boldsymbol{A}^T\boldsymbol{A}$ decides on the sensitivity of the problem. The smaller the condition number the better. For numerical robustness it is also important that the smallest singular value of this matrix is well above machine precision. We make use of this observation to state an algorithm which allows the automatic estimation of the best scaling factor in terms of a minimum condition number. First we translate the points such that their centroid is the origin of the coordinate system. Then the scaling of points is computed by solving an one–dimensional optimization task. The condition number $\kappa(\boldsymbol{A}^T\boldsymbol{A})$ can be implemented as a function of scaled calibration points. Thus the optimal scale factor in terms of numerical robustness is given by:

$$\widehat{s} = \operatorname{argmin}_s \kappa(\boldsymbol{A}^T\boldsymbol{A}) \ . \tag{3}$$

4.3 On the Design of Optimal Calibration Patterns

The introduced parameter estimation algorithm is a linear transform of input data. If the covariance matrix of measurements is $\boldsymbol{\Sigma}$ then the covariance of estimates is $\boldsymbol{A}^{\dagger}\boldsymbol{\Sigma}(\boldsymbol{A}^{\dagger})^T$. In many applications, however, the mapping from measurements to parameter vectors is not closed form but defined by an optimization problem. In [3] the authors introduce formulas to compute estimates of covariance matrices even in the presence of constraints on parameters.

The covariance matrix of estimates allows the analysis of the sensitivity of applied algorithm. We get concrete numbers which quantify both the statistical dependencies as well as the accuracy of each estimate. Besides the applied cost function and the selected parameterization, the sensitivity of estimated parameters also depends on the spatial distribution of calibration points. It is intuitive and obvious that the sampling density has to be higher, the more the sampled function varies. Less variations require less sampling points. The design of a calibration pattern, which is optimal in the sense of low variances in estimates, is guided by a combinatorial search algorithm. An appropriate cost function which judges each configuration of calibration points is the determinant of the covariance matrix $\boldsymbol{\Sigma}$ of estimates. For a given number of calibration points we select the configuration which leads to the lowest determinant:

$$(\widehat{x}_n, \widehat{y}_n)_{n=1,2,\dots,N} = \operatorname{argmin}_{(x_n,y_n)} \det(\boldsymbol{\Sigma}) \ . \tag{4}$$

5 Efficient Evaluation of Polynomials

After the polynomial mapping is computed efficient algorithms are required for the evaluation of warp functions. In case of a bivariate polynomials we simply consider the polynomial as a polynomial in x where the coefficients are polynomials in y (see (1)). Both coefficients and the polynomial in x can be evaluated in an efficient and robust manner by Horner's rule. If we consider the problem of image warping we notice that the coefficients of the coefficents of x are constant for fixed y. Thus the polynomials in y are evaluated once for each image row.

Another restriction is that polynomials are evaluated for image grid points only. Increments in x– and y–direction remain constant during the whole warp process. Therefore we can apply the tabulating algorithm suggested in [4]. It allows the efficient evaluation of a polynomial using arithmetic progression. After initialization the evaluation of polynomials is reduced to a few additions only. This idea can be applied to the polynomial in x and to its coefficients.

6 Summary

This paper considered the problem of image undistortion which is important for many reconstruction algorithms. We described various facets of this problem from a computational point of view including the definition of cost functions, the judgment of estimates and the design of optimal calibration patterns. The introduced concepts of stability, fairness, and scaling are important for the sensitivity analysis of parameter estimation algorithms and necessary to study intrinsic properties.

References

1. Hartley R, Zissermann A: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, 2000
2. Lehmann TE, Goenner C, Spitzer K: Survey: Interpolation Methods in Medical Image Processing. IEEE Trans. on Medical Imaging, 18(11):1049–1075, 1999.
3. Hornegger J, Tomasi C: Representation Issues in the ML Estimation of Camera Motion. In Proceedings of Seventh International Conference on Computer Vision, Kerkyra, pp. 640–647, 1999
4. Knuth DE: The Art of Computer Programming, Volume 2, Seminumerical Algorithms. Addison–Weseley, Reading, MA, Third Edition, 1997.