ON AUTOMATIC DETERMINATION OF VARYING FOCAL LENGTHS USING SEMIDEFINITE PROGRAMMING

Motilal Agrawal

SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
agrawal@ai.sri.com

ABSTRACT

We describe a novel approach to the determination of the focal length of a moving camera with rectangular pixels. The principal point of the camera is assumed to be known and fixed, whereas the focal length is allowed to vary across the sequence. Given three or more such images and a projective reconstruction, we describe a novel auto-calibration technique to obtain a metric reconstruction. Our technique uses semidefinite programming to recover these focal lengths (and hence the metric reconstruction). Our approach is efficient, well behaved with guaranteed convergence, and can be applied to long sequences of video. We present results for our approach using both simulated and real video sequences.

1. INTRODUCTION

Reconstruction of $3D$ scenes from a video sequence is a fundamental problem of computer vision with wide applicability. Over the last few years, digital video cameras have become ubiquitous because of their declining costs, thereby giving everybody easy access to video sequences. However, processing of these video sequences to reconstruct $3D$ scenes has still not reached the commercialization stage. This can happen by advancing the state of the art in *uncalibrated* structure from motion.

In the uncalibrated setting, the best one can achieve is a reconstruction of the scene up to an unknown projective transformation. This projective structure, however, is not good enough for many applications. To obtain a euclidean reconstruction, knowledge about the calibration parameters of the camera is required. The widely used approach is to employ a calibration pattern to acquire these parameters offline. This is, however, too restrictive. Auto-calibration, on the other hand, recovers these parameters from an unknown scene using the rigidity constraints present in the scene and certain simplifying assumptions about the camera (e.g., rectangular/square pixels, known principal points, constant intrinsic parameters).

In this paper, we describe a novel aplication of semidefinite programming for focal length recovery. Our formulation is based on the assumption that the camera has rectangular pixels (skewless camera) and the principal point of the camera is known and fixed. Hence, the variables to be estimated are the aspect ratio and the focal length of the camera at each frame of the sequence. The principal point of the camera is known to be notoriously hard to compute. Often, therefore, the principal point position is guessed to be at the center of the image. In addition, the principal point is known to vary with the zoom setting of a camera (several authors have reported the principal point to move as much as 30 pixels with the zoom [1]). However, our experiments, using our approach indicate that the inaccuaracies in the (guessed) principal point do not have a significant effect on the other parameters of the camera. Using these assumptions, we formulate the auto-calibration problem in the semidefinite programming framework (section 4). Results using our approach for a simulated and real sequence are presented in section 5. A brief introduction to semidefinite programming is provided in section 3. We begin by describing the auto-calibration problem and the various approaches for solving it.

2. BACKGROUND

We assume that the reader is familiar with $3D$ computer vision. Therefore, because of space restrictions, standard widely used facts will be stated without derivation. Please refer to [2] for details and proofs. The basic projection equation for a camera with projection matrix $P^i = \begin{bmatrix} A^i & a^i \end{bmatrix}$ is given by $x_j = P^i X_j$. Here, X_j is the homogenous coordinate of the $3D$ point, and x_j is its projection in the image. P^i contains information about the pose of the camera and the intrinsic parameters, represented by the matrix $K^i = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$. Here, α_x, α_y are the focal lengths in pixels in the x and y directions, s is the camera skew, and x_0, y_0 are the principal points of the camera. The absolute conic is an imaginary conic closely tied to the intrinsic parameters. Its dual (called the DIAC) is represented by $\omega^{*i} = K^i K^{it}$, where K^{it} is the transpose of K^i. It is easy to see that the matrix ω^{*i} is symmetric and positive semidefinite (denoted by $\omega^{*i} \succeq 0$). If the skew of the camera is zero ($s = 0$), then ω^{*i} can be written compactly as

$$\omega^{*i} = F_0 + \alpha_x^2 F_1 + \alpha_y^2 F_2 \quad (1)$$

$$F_0 = \begin{bmatrix} x_0^2 & x_0y_0 & x_0 \\ x_0y_0 & y_0^2 & y_0 \\ x_0 & y_0 & 1 \end{bmatrix} \quad (2)$$

$$F_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

$$F_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4)$$

The goal of auto-calibration is to recover the DIAC by exploiting the rigidity constraints present in the scene. The intrinsic parameters (K^i) can then be recovered from ω^* by Cholesky factorization, thereby updating the projective structure to a metric one.

The earliest approach to auto-calibration involved the use of Kruppa's equation [3]. These are two view constraints that require only the fundamental matrix between the two views to be known and consist of two independent quadratic equations in the elements of ω^*. Although these are the only constraints available for two views, application to three or more views does not yield satisfactory results.

The stratified approach to auto-calibration [4] first recovers the plane at infinity (π_∞), thereby obtaining an affine reconstruction, and then subsequently recovers the intrinsic parameter matrix. The difficulty of this approach lies in estimating π_∞. Methods that use scene geometry identify vanishing points of lines to determine π_∞. However, they do not fall strictly in the regime of auto-calibration. If the internal parameters of the camera remain constant, then given three views, π_∞ can be obtained as the intersection of three quartic equations in three variables. This is known as the *modulus constraint*. Because this approach requires the solution of simultaneous sets of quartic equations, very often it does not yield satisfactory results. In fact, the modulus constraint is closely related to Kruppa's constraints. Kruppa's constraints eliminates π_∞ yielding a constraint on ω^*, whereas the modulus constraint eliminates ω^* yielding an equation in π_∞. Therefore, these two methods perform similarly.

The third and perhaps the most widely used approach for a large number of views solves for both ω^* and π_∞ simultaneously. This approach was introduced by Triggs [5] in terms of an equivalent formulation using the absolute quadric - an imaginary degenerate quadric represented by a 4×4 matrix of rank 3 (Q_∞). Given n cameras, the equations relating the unknown parameters ω^{i*} and π_∞ with the known entries of the projection matrices P^i, $i = 1, \ldots, n$ are given as

$$\kappa^i \omega^{*i} = \left(A^i - a^i \pi_\infty^t\right) \omega^{*1} \left(A^i - a^i \pi_\infty^t\right)^T \quad \text{i=2,\ldots,n} \quad (5)$$

$$= P^i Q_\infty^* P^{iT} \quad (6)$$

$$Q_\infty^* = \begin{bmatrix} \omega^{*1} & -\omega^{*1}\pi_\infty \\ -\pi_\infty \omega^{*1} & \pi_\infty \omega^{*1} \pi_\infty \end{bmatrix} \quad (7)$$

Here κ^i is an unknown scale factor. Since $\omega^{*1}, \omega^{*i} \succeq 0$, it is easy to see that the RHS is SDP and $\kappa^i \geq 0$. Using eqn. 5, constraints on K^i are translated into constraints on ω^i that in turn gives an equation relating ω^{*1} and π_∞. Given enough such constraints, it is possible to solve for these unknowns. When the intrinsic parameters of the camera remain constant, Triggs used a non-linear constrained minimization approach. The results of this approach are, however, strongly influenced by the initial solution. For a skewless camera with known principal point, these constraints become linear and therefore can be solved for using linear least squares (LLS) [6]. However, this approach has two major drawbacks.

1. The rank 3 constraint on Q_∞^* is not enforced in the solution. This can be later enforced in a posteriori step wherein the closest rank 3 matrix of Q_∞^* is used as the starting point for an iterative minimization step. However, the results are not guaranteed.

2. The most troublesome failing is that the positive definiteness condition of ω^* is completely ignored in the optimization step. As a result, we may end up with a ω^* that is *not* positive definite, and thus the calibration parameters cannot be recovered. The closest positive semidefinite matrix to the computed ω^* will, in most cases, not be the correct one. Hence, we have to reject this solution and start over. This situation becomes all the more prominent because of the presence of noise in the data.

Our approach overcomes these two drawbacks of the linear algorithm in a very natural manner using semidefinite programming. In practice, the solution from the linear algorithm is often used as the initial point for a non-linear approach. Our approach provides an alternative method for initialization with a guaranteed convergence and a better starting solution.

3. SEMIDEFINITE PROGRAMMING

Semidefinite programming (denoted SDP) is an extension of linear programming (LP) where the nonnegativity constraints are replaced by positive semidefiniteness constraints on matrix variables. SDP is a very powerful tool that has found applications in positive definite completion problems, maximum entropy estimation and bounds for hard combinatorial problems; see, for example, the survey of Vandenberghe and Boyd [7]. Recently, SDP has also been used to calibrate a camera using spheres [8].

The standard dual form of SDP can be expressed as minimizing a linear function of a variable $x \in \mathcal{R}^m$ subject to a matrix inequality

$$\begin{array}{ll} \text{minimize} & c^t x \\ \text{subject to} & F(x) \succeq 0 \\ F(x) = F_0 + & \sum_{i=1}^m x_i F_i \end{array} \tag{8}$$

The problem data are the vector $c \in \mathcal{R}^m$ and $m+1$ symmetric matrices $F_0, \ldots, F_m \in \mathcal{R}^{n \times n}$. The inequality sign in $F(x) \succeq 0$ means that $F(x)$ is symmetric positive definite.

Unlike LP, SDP is a nonlinear convex programming problem, because the feasible boundary (the cone of positive semidefinite matrices) is nonlinear. Nonetheless, SDP shares a key feature with LP — SDP can be effectively solved by generalizing interior-point methods developed originally for LP. The credit for this important discovery goes primarily to Nesterov and Nemirovski [9]. The fact that SDP can be efficiently solved by interior–point methods, both in practice and, from a theoretical point of view, *exactly* in polynomial time, has driven the recent surge of interest in the subject.

3.1. Norm minimization using SDP

Suppose a matrix $A(x)$ depends affinely on $x \in \mathcal{R}^k : A(x) = A_0 + x_1 A_1 + \cdots + x_k A_k \in \mathcal{R}^{p \times q}$, and we want to minimize the spectral norm (maximal singular value) $\|A(x)\|$. This can be cast as the semidefinite program.

$$\begin{array}{ll} \text{minimize} & \zeta \\ \text{subject to} & M = \begin{bmatrix} \zeta I & A(x) \\ A(x)^t & \zeta I \end{bmatrix} \succeq 0 \end{array} \tag{9}$$

If, in addition, we have a constraint that the matrix $C(x) = C_0 + x_1 C_1 + \cdots + x_k C_k \succeq 0$, then this can be incorporated by replacing the constraint in eqn. 9 with

$$\text{diag}(M, C(x)) \succeq 0 \tag{10}$$

Here, diag(A,B) is the block diagonal matrix with diagonal entries A and B. If $B(x)$ is another matrix that depends linearly on x and we want to minimize $\|A(x)\| + \|B(x)\|$, then the corresponding semidefinite program to be solved is

$$\begin{array}{ll} \text{minimize} & \zeta + \eta \\ \text{subject to} & \text{diag}\left(\begin{bmatrix} \zeta I & A(x) \\ A(x)^t & \zeta I \end{bmatrix}, \begin{bmatrix} \eta I & B(x) \\ B(x)^t & \eta I \end{bmatrix} \right) \succeq 0 \end{array} \tag{11}$$

As before, it is easy to add constraints such as $C(x) \succeq 0$ in this framework. Generealizing this, it is easy to see that SDP can be used to minimize the sum of norms of arbitrary matrices that are affinely dependent on a variable subject to SDP constraints. In the next section, we illustrate how to formulate the auto-calibration problem for any number of given cameras into a norm minimization problem.

4. AUTO-CALIBRATION USING SDP

We present our SDP-based formulation of the auto-calibration of focal lengths for a skewless camera with known principal point. We are given n images of a rigid scene obtained by a moving camera with variable focal lengths. Feature points can then be matched across these images, and a projective reconstruction with projection matrices P_i, $i = 1, \ldots, n$ can be obtained [10]. Our goal is to recover α_x^i and α_y^i for each camera. Substituting the expression for ω^* (eqn.1) into the basic equation for auto-calibration (eqn.5) for the i^{th} image, we obtain

$$\begin{aligned} \kappa^i \left(F_0 + \alpha_x^{i2} F_1 + \alpha_y^{i2} F_2\right) \quad = \quad & \left(A^i - a^i \pi_\infty^t\right)\left(F_0 + \alpha_x^2 F_1\right. \\ & \left. + \alpha_y^2 F_2\right)\left(A^i - a^i \pi_\infty^t\right)^T \end{aligned} \tag{12}$$

Let $\pi_\infty^t = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}$ and e_1, e_2, e_3 be the three standard basis vectors for the group of 3×3 matrices (i.e., $e_1^t = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ etc;). Let $\gamma_7^i = \kappa^i$, $\gamma_8^i = \kappa^i \alpha_x^{i2}$ and $\gamma_9^i = \kappa^i \alpha_y^{i2}$. Then LHS can be written as

$$LHS^i = \sum_{j=0}^{2} \gamma_{7+j}^i F_j \tag{13}$$

After multiplying the terms, the RHS can be written as an affine combination of seven symmetric matrices $G_0^i, \ldots, G_6^i$.

$$\begin{aligned} RHS^i \quad = & A^i F_0 A^{iT} + \alpha_x^2 A^i F_1 A^{iT} + \alpha_y^2 A^i F_2 A^{iT} + \\ & \alpha_x^2 n_1 \left(A^i e_1 a^{iT} + a^i e_1^T A^{iT}\right) + \\ & \alpha_y^2 n_1 \left(A^i e_2 a^{iT} + a^i e_2^T A^{iT}\right) + \\ & n_3 \left(A^i e_3 a^{iT} + a^i e_3^T A^{iT}\right) + \\ & \left(n_3^2 + \alpha_x^2 n_1^2 + \alpha_y^2 n_2^2\right) a^i a^{iT} \end{aligned} \tag{14}$$

Let $\gamma_1 = \alpha_x^2$, $\gamma_2 = \alpha_y^2$, $\gamma_3 = \alpha_x^2 n_1$, $\gamma_4 = \alpha_y^2 n_2$, $\gamma_5 = n_3$ and $\gamma_6 = n_3^2 + \alpha_x^2 n_1^2 + \alpha_y^2 n_2^2$. Thus

$$RHS^i = G_0^i + \sum_{j=1}^{6} \gamma_j G_j^i \tag{15}$$

Therefore, the autocalibration problem can be cast as the minimization of the sum of norm of $n-1$ matrices subject to certain SDP constraints.

$$\text{minimize} \sum_{i=2}^{n} \|\text{LHS}^i - \text{RHS}^i\| \tag{16}$$

4.1. Constraints

Because of the parameterization that we are using, the rank constraint for Q_∞^* is automatically enforced. It is also easy to add the constraint that ω_i^* is positive semidefinite.

$$F_0 + \gamma_1 F_1 + \gamma_2 F_2 \succeq 0 \tag{17}$$

$$\gamma_7^i F_0 + \gamma_8^i F_1 + \gamma_9^i F_2 \succeq 0 \quad i = 2, \ldots, n \tag{18}$$

The expressions for γ_1, γ_2 and γ_6 imply that all of these are non-negative. Similarly, γ_7^i, γ_8^i and γ_9^i are also non-negative. Therefore, we have

$$\text{diag}\left(\gamma_1, \gamma_2, \gamma_6, \gamma_7^2, \ldots, \gamma_7^n, \gamma_8^2, \ldots, \gamma_8^n, \gamma_9^2, \ldots, \gamma_9^n\right) \succeq 0 \tag{19}$$

The above SDP constraints along with the norm minimization of the sum in eqn. 16 make a constrained norm minimization problem. The variables in this case are $\gamma_1, \ldots, \gamma_6, \gamma_7^i, \gamma_8^i, \gamma_9^i$ for $i = 2, \ldots, n$. SDP can then be used to solve this minimization problem and obtain the gammas from which the focal length of each camera can then be obtained.

5. RESULTS

We have applied our algorithm for both simulated data and real images. For the simulations, a random cloud of 500 $3D$ points was generated. Images of these points were obtained in a sequence of 10 frames by moving the camera and changing its focal lengths while keeping the principal point fixed. Random gaussian noise was then added to the location of the projected points. Using these point correspondences, a projective reconstruction was obtained using the iterative factorization algorithm of [10]. Finally, the projective reconstruction was upgraded to a metric one using our algorithm. Table 1 shows the recovered focal lengths for frames 1, 6 and 10 using our algorithm(SDP) for noise with $\sigma = 0.5, 1.0, 2.0$ pixels ($\sigma = 0$ corresponds to ground truth). The running time was a couple of seconds. For comparison, the results of the linear algorithm (lin) are also included and the true focal lengths are given in the third column. NaN indicates that the focal lengths could not be recovered. Clearly, our algorithm performs much better than the linear algorithm. The maximum relative error in the recovered focal lengths for noise of 0.5, 1.0 and 2.0 was 3%, 6.8% and 7.5%, respectively.

Table 1. Results for simulated data

Cam	$\sigma = 0$	$\sigma = 0.5$		$\sigma = 1.0$		$\sigma = 2.0$	
		lin	SDP	lin	SDP	lin	SDP
α_x^1	666	414	650	78	688	NaN	670
α_y^1	673	479	663	150	641	266	622
α_x^6	629	466	618	130	663	93	658
α_y^6	633	510	630	116	622	250	604
α_x^{10}	603	534	584	52	644	NaN	648
α_y^{10}	649	598	642	74	628	240	626

In our second experiment, a camera with fixed parameters was moved in a rigid indoor scene. Thirty feature points were obtained and tracked in ten frames of this sequence. A projective reconstruction was obtained using these correspondences, and our algorithm was applied to obtain the camera parameters. Since the camera parameters were fixed, the recovered focal lengths should be constant. However, due to noise these focal lengths will vary. The mean focal lengths obtained for these 10 frames was $\alpha_x = 911$ and $\alpha_y = 917$. The standard deviation for the focal lengths was $\mathrm{sd}_x = 3.1$ and $\mathrm{sd}_y = 3.6$, indicating that recovered focal lengths did not vary much (the center of the image was taken as the principal point). The camera was also calibrated offline using a plane calibration grid yielding focal lengths $\alpha_x = 928$ and $\alpha_y = 926$, thus verifying the recovered focal lengths (The focal length obtained by the linear algorithm was 103 and 117.)

6. CONCLUSION

We have presented a novel approach the recovery of focal lengths using semidefinite programming. Although our approach assumes that the principal point of the camera is known, it has been our experience that this point can often be assumed to be at the center of the image without much impact on the computed focal lengths. Our approach naturally incorporates the constraint that ω^* is positive semidefinite and therefore is much more stable. In addition, it is also efficient and can incorporate a large number of views easily. We have presented experimental results with both synthetic and real sequence. We are currently looking into modifying our algorithm to account for the fact that the principal point is unknown. In cases where the principal point varies substantially with the zoom setting of the camera, we can apply our approach to shorter segments wherein the focal length is known to be relatively constant.

Acknowledgement: Financial support from the US Army project #DAAD19-01-2-0012 under subcontract #9613P with General Dynamics Robotic Systems is gratefully acknowledged.

7. REFERENCES

[1] R.Collins and Y.Tsin, "Calibration of an outdoor active camera system," in *Proc. CVPR*, 1999, pp. 528–534.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[3] Q. Luong and O. Faugeras, "Self-calibration of a moving camera from point correspondences and fundamental matrices," *IJCV*, vol. 22, no. 3, pp. 261–289, 1997.

[4] M. Pollefeys and L. Van Gool, "A stratified approach to metric self-calibration," in *Proc. CVPR*, 1997, pp. 407–412.

[5] B. Triggs, "Autocalibration and the absolute quadric," in *Proc. CVPR*, 1997, pp. 609–614.

[6] M. Pollefeys, R. Koch, and L. van Gool, "Self-calibration and metric reconstruction inspite of varying and unknown internal camera parameters," in *Proc. ICCV*, 1998, pp. 90–95.

[7] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[8] M. Agrawal and L. Davis, "Camera calibration using spheres: A semidefinite programming approach," in *ICCV*, 2003, pp. 782–789.

[9] Y.E. Nesterov and A.S. Nemirovski, *Interior point polynomial algorithms in convex programming*, SIAM publications, Philadelphia, USA, 1994.

[10] S. Mahamud and M. Hebert, "Iterative projective reconstruction from multiple views," in *Proc. CVPR Vol. II*, 2000.