

MODELLAZIONE, STIMA E RIMOZIONE DELLA DISTORSIONE RADIALE DA UN'IMMAGINE

Carlo Colombo

— ultima revisione: 19 ottobre 2005 —

Modellazione (e Rimozione)

La distorsione radiale è legata all'uso di lenti di non buona qualità e/o all'esigenza di ampliare il campo visivo utilizzando una telecamera con distanza focale corta. L'effetto della distorsione è quello di far apparire curve le linee rette presenti nella scena. L'entità della distorsione è tanto maggiore quanto più ci si allontana dalla parte centrale dell'immagine [2].

Detta f la distanza focale, l'immagine di un punto P in coordinate “ideali” è

$$(x, y) = f({}^cX/{}^cZ, {}^cY/{}^cZ) , \quad (1)$$

dove $({}^cX, {}^cY, {}^cZ)$ rappresenta il punto P in coordinate di telecamera, e il punto proiettato (x, y) è espresso nelle stesse unità di misura di f (ad es. pixel, o mm). La distorsione radiale è introdotta dalla lente *prima* dell'acquisizione digitale dell'immagine. Il punto in coordinate “ideali” viene distorto secondo la legge

$$(\check{x}, \check{y}) = (x, y) \delta(r) , \quad (2)$$

dove $r = \sqrt{x^2 + y^2}$, e la *funzione di distorsione* $\delta(r)$ è non negativa e crescente con r , avendosi $\delta(0) = 1$ (no distorsione per $x = y = 0$). Indicata con $\check{r} = \sqrt{\check{x}^2 + \check{y}^2}$ la distanza del punto distorto rispetto all'origine, si può scrivere $\check{r} = r \delta(r)$. Tradizionalmente, la distorsione radiale non è modellata attraverso la funzione $\delta(\cdot)$, ma attraverso la sua inversa $\delta^{-1}(\cdot)$ tale che

$$(\check{x}, \check{y}) \delta^{-1}(\check{r}) = (x, y) ; \quad \check{r} \delta^{-1}(\check{r}) = r . \quad (3)$$

La funzione di distorsione inversa più comunemente utilizzata è di tipo polinomiale:

$$\delta^{-1}(\check{r}) = 1 + k_1 \check{r}^2 + k_2 \check{r}^4 + \dots , \quad (4)$$

dove k_i indica il coefficiente di distorsione i -simo. In particolare, per modellare la distorsione radiale dovuta a lenti di qualità accettabile è sufficiente considerare solo il primo coefficiente; la legge di distorsione risultante è di tipo quadratico:

$$\delta^{-1}(\check{r}) = 1 + k_1 \check{r}^2 . \quad (5)$$

Il coefficiente k_1 può assumere in generale valori sia positivi (distorsione di tipo *barrel*, o “a botte”) che negativi (distorsione di tipo *pin cushion*, o “a cuscino”). Nella pratica della computer vision il caso più comune è quello di $k_1 > 0$ [1].

A parte la distorsione di tipo quadratico, la lente ne introduce anche una di tipo lineare:

$$\begin{aligned}\check{x}' &= \sigma \check{x} + q_x, \\ \check{y}' &= \check{y} + q_y,\end{aligned}\tag{6}$$

dove σ (adimensionale) modella una distorsione di tipo tangenziale, e (q_x, q_y) — in coordinate “ideali” — rappresenta una traslazione rispetto all’origine del centro di distorsione. Il punto così ottenuto viene infine digitalizzato, subendo un’ulteriore deformazione di tipo lineare dovuta ai parametri interni della telecamera (modello qui utilizzato: a pixel rettangolari, o a “skew” nullo):

$$\begin{aligned}\check{x}'' &= \rho \check{x}' + p_x, \\ \check{y}'' &= \check{y}' + p_y,\end{aligned}\tag{7}$$

dove ρ è il cosiddetto *aspect ratio* (rapporto tra il lato orizzontale e quello verticale del pixel), e (p_x, p_y) sono le coordinate (in pixel) del *punto principale* (proiezione sull’immagine del centro ottico della telecamera). Per effetto di quest’ultima distorsione lineare, il punto in coordinate “ideali” (\check{x}', \check{y}') viene espresso nelle coordinate “non ideali” $(\check{x}'', \check{y}'')$, dette anche *coordinate immagine*. La parte lineare della distorsione è dunque data dalla composizione delle due distorsioni lineari — quella dovuta alla lente, e quella dovuta alla digitalizzazione:

$$\begin{aligned}\check{x}'' &= \tau \check{x} + r_x, \\ \check{y}'' &= \check{y} + r_y,\end{aligned}\tag{8}$$

con $\tau = \rho \sigma$, $r_x = p_x + \rho q_x$, e $r_y = p_y + q_y$. In genere τ è un numero vicino a 1, mentre (r_x, r_y) è vicino al centro immagine espresso in pixel, e si ha $p_x \gg q_x$, $p_y \gg q_y$, cosicché $(r_x, r_y) \approx (p_x, p_y)$. Invertendo la relazione precedente, otteniamo

$$\begin{aligned}\check{x} &= (\check{x}'' - r_x)/\tau, \\ \check{y} &= \check{y}'' - r_y,\end{aligned}\tag{9}$$

da cui

$$\begin{aligned}x &= \check{x}(1 + k_1 \check{r}^2) = \left(\frac{\check{x}'' - r_x}{\tau}\right) \left\{1 + k_1 \left[\left(\frac{\check{x}'' - r_x}{\tau}\right)^2 + (\check{y}'' - r_y)^2\right]\right\}, \\ y &= \check{y}(1 + k_1 \check{r}^2) = (\check{y}'' - r_y) \left\{1 + k_1 \left[\left(\frac{\check{x}'' - r_x}{\tau}\right)^2 + (\check{y}'' - r_y)^2\right]\right\}.\end{aligned}\tag{10}$$

Si ottiene infine

$$\begin{aligned}x'' &= \tau x + r_x = \check{x}'' + (\check{x}'' - r_x) k_1 \left[\left(\frac{\check{x}'' - r_x}{\tau}\right)^2 + (\check{y}'' - r_y)^2\right], \\ y'' &= y + r_y = \check{y}'' + (\check{y}'' - r_y) k_1 \left[\left(\frac{\check{x}'' - r_x}{\tau}\right)^2 + (\check{y}'' - r_y)^2\right],\end{aligned}\tag{11}$$

relazione che esprime un punto in coordinate immagine prive di distorsione radiale $\mathbf{x} = (x'', y'')$ in funzione del punto in coordinate immagine con distorsione radiale $\check{\mathbf{x}} = (\check{x}'', \check{y}'')$ osservabile sull’immagine e dei parametri di acquisizione k_1 , τ , r_x e r_y . Una volta stimati questi ultimi — si noti che ciò non comporta né implica la stima dei parametri di calibrazione —, è possibile usare l’eq. (11) per trasformare i punti distorti osservati sull’immagine nei corrispondenti punti privi di distorsione.

Stima

I parametri che compaiono nell'equazione non lineare (11) possono essere stimati con un metodo iterativo, ad es. quello di *Levenberg-Marquardt* [2].

Si supponga di conoscere, per averli calcolati sull'immagine distorta, due punti $\hat{\mathbf{x}}_0 = (\hat{x}_0'', \hat{y}_0'')$ e $\hat{\mathbf{x}}_N = (\hat{x}_N'', \hat{y}_N'')$, situati agli estremi di un segmento di curva che è rettilineo nella scena (e tale sarebbe nell'immagine, in assenza di distorsione radiale). Si supponga inoltre di aver calcolato un insieme di punti intermedi $\hat{\mathbf{x}}_i = (\hat{x}_i'', \hat{y}_i'')$ ($i = 1 \dots N-1$) appartenenti allo stesso segmento. Per effetto del rumore di misura, i punti non giacciono *esattamente* sul segmento, ma si trovano nei suoi pressi (Figura 1). Tracciata la *retta di test* di equazione $a\hat{x}'' + b\hat{y}'' + c = 0$ passante per $\hat{\mathbf{x}}_0$ e $\hat{\mathbf{x}}_N$, la distanza del generico punto intermedio $\hat{\mathbf{x}}_i$ da tale retta è

$$\varepsilon_i = \frac{|a\hat{x}_i'' + b\hat{y}_i'' + c|}{\sqrt{a^2 + b^2}}, \quad (12)$$

con $a = \hat{y}_0'' - \hat{y}_N''$, $b = \hat{x}_N'' - \hat{x}_0''$, e $c = \hat{x}_0''\hat{y}_N'' - \hat{x}_N''\hat{y}_0''$.

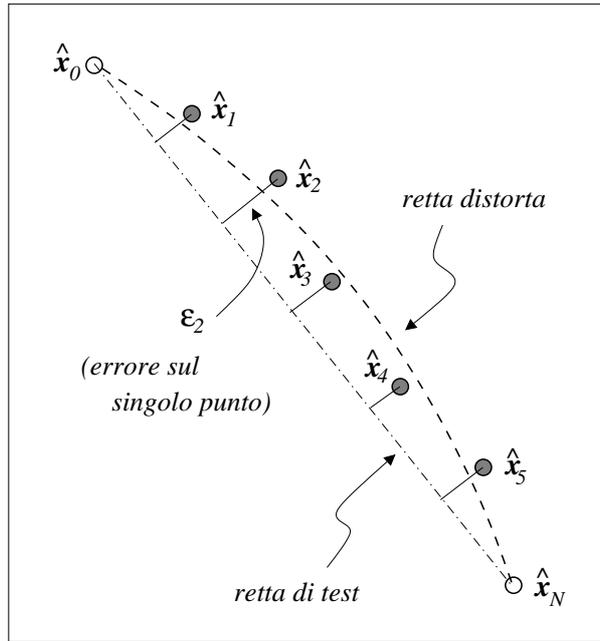


Figura 1: Distorsione radiale e misure sull'immagine.

Le implementazioni dell'algoritmo di Levenberg-Marquardt reperibili in rete (si veda ad es. [3]) richiedono essenzialmente (i) un valore di inizializzazione per ciascuno dei parametri da stimare; (ii) un insieme di osservazioni; (iii) una funzione errore calcolabile per ciascuna osservazione. Nel caso della stima della distorsione radiale, (i) i valori di inizializzazione dei parametri possono scegliersi come segue: $k_1 = 0$, $\tau = 1$, $r_x = w/2$, $r_y = H/2$, con w e H rispettivamente la larghezza (width) e l'altezza (height) dell'immagine; (ii) le osservazioni sono costituite da insiemi di punti del tipo $\{\hat{\mathbf{x}}_i^\ell\}_{i=0}^{N_\ell}$, uno per ciascuna retta ℓ considerata nella scena; (iii) la funzione errore è quella che consente di calcolare per ogni $\hat{\mathbf{x}}_i^\ell$ un punto $\hat{\mathbf{x}}_i^\ell$ corrispondente, e quindi la sua distanza dalla relativa retta di test (a_ℓ, b_ℓ, c_ℓ).

In dettaglio, il funzionamento dell'algoritmo di minimizzazione non lineare che consente la stima dei parametri di distorsione può riassumersi come segue:

0. Poni $t = 0$ (t è un contatore che tiene traccia del numero di iterazioni già eseguite). Imposta il valore iniziale dei parametri $k_1(0)$, $\tau(0)$, $r_x(0)$ e $r_y(0)$ secondo quanto detto sopra.
1. Usa i parametri stimati $k_1(t)$, $\tau(t)$, $r_x(t)$ e $r_y(t)$, e le osservazioni $\left\{ \left\{ \tilde{\mathbf{x}}_i^\ell \right\}_{i=0}^{N_\ell} \right\}_{\ell=1}^{\mathcal{L}}$ per rimuovere la distorsione secondo l'eq. (11), ottenendo così l'insieme di osservazioni "corrette" $\left\{ \left\{ \hat{\mathbf{x}}_i^\ell(t) \right\}_{i=0}^{N_\ell} \right\}_{\ell=1}^{\mathcal{L}}$.
2. Costruisci l'insieme di rette $\left\{ (a_\ell(t), b_\ell(t), c_\ell(t)) \right\}_{\ell=1}^{\mathcal{L}}$ usando i punti estremi $\hat{\mathbf{x}}_0^\ell(t)$ e $\hat{\mathbf{x}}_{N_\ell}^\ell(t)$ di ogni retta. Usa l'eq. (12) per calcolare l'insieme degli errori $\left\{ \left\{ \varepsilon_i^\ell(t) \right\}_{i=0}^{N_\ell} \right\}_{\ell=1}^{\mathcal{L}}$. Se l'errore complessivo è inferiore alla soglia ϑ , allora termina (con successo).
3. Incrementa t ; se $t > t_{\max}$ allora termina (con fallimento), altrimenti aggiorna la stima dei parametri con il metodo di Levenberg-Marquardt, e torna a 1.

In pratica, le versioni C/C++ dell'algoritmo disponibili in rete consentono di ottenere una stima dei parametri a partire dall'input (valori di inizializzazione, osservazioni sull'immagine, funzione errore, soglie sul numero di iterazioni e sull'errore, etc.) con una semplice chiamata a funzione, del tipo

```
exit_code = LM(parametri,funzione_errore,osservazioni,soglie);
```

La funzione di errore va programmata ad hoc, e passata alla funzione LM tipicamente attraverso un puntatore a funzione.

Riferimenti bibliografici

- [1] F. Devernay, O. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, Vol. 13, pp. 14-24, 2001.
- [2] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003².
- [3] M.I.A. Lourakis, "levmar: Levenberg-Marquardt non-linear least squares algorithms in C/C++," www.ics.forth.gr/~lourakis/levmar, 2004.