



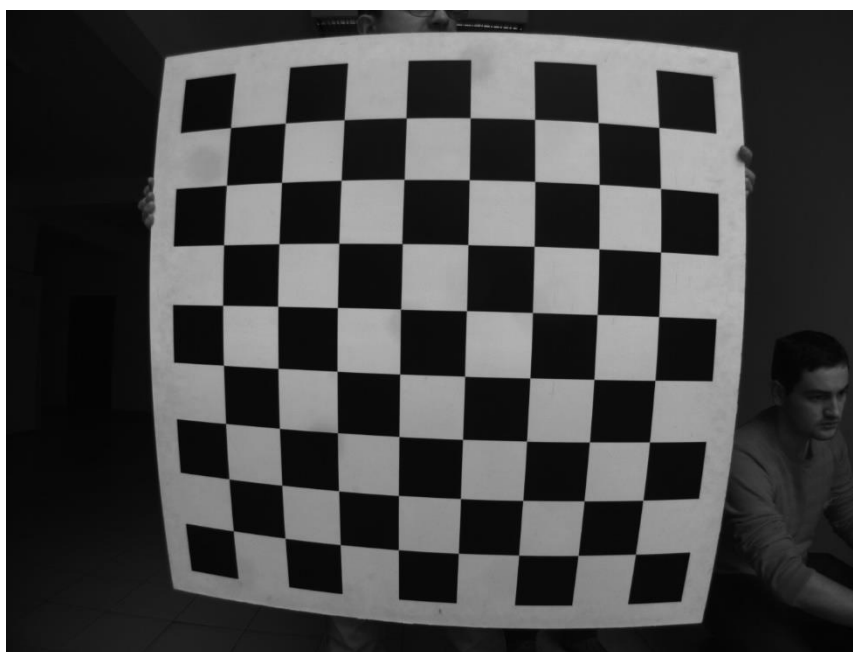
**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

## **Documentație proiect**

### **Eliminarea distorsiunilor lentilelor**



Profesor îndrumător: Vancea Cristian

Student: Astaliș Lorena-Maria

Grupa: 30234

Anul 2022

# Cuprins

1. *Introducere*
2. *Considerații teoretice*
3. *Specificații de implementare*
4. *Rezultate experimentale*
5. *Concluzii*
6. *Bibliografie*

# 1. Introducere

Obiectivul acestei lucrări este de a elimina distorsiunile din imagini generate de lentilele unei camere (se cunosc parametrii intrinseci ai acesteia). Ideal, o fotografie are o perspectivă perfect mapată pe scena reală care este surprinsă de aceasta, în realitate, mai ales când dorim să folosim imagini pentru a reconstrui un accident sau alte situații în care avem nevoie de o reprezentare cât mai realistă pe imagini pentru a deduce diverse date, acestea nu oferă informații relevante în forma brută care ajung, din cauza lentilelor folosite de către cameră. Pentru a ține distorsiunile lentilelor la un nivel cât mai scăzut se recomandă un obiectiv cu unghi mic. În practică acest lucru este greu de obținut din cauza distanței față de care trebuie fotografiat obiectul, pentru cazul cu reconstrucția unui accident de obicei avem nevoie de imagini cu un câmp vizual larg. Din fericire prin tehnici software putem elimina aceste distorsiuni ale imaginilor pentru a elimina cât mai mult aceste erori. [1]

## 2. Considerații teoretice

Ideea generală de la care se pornește în implementarea unei soluții de eliminare a distorsiunilor unei imagini constă în maparea pixelilor din imaginea cu distorsiune în imaginea rezultat care elimină curbura generată de obiective.

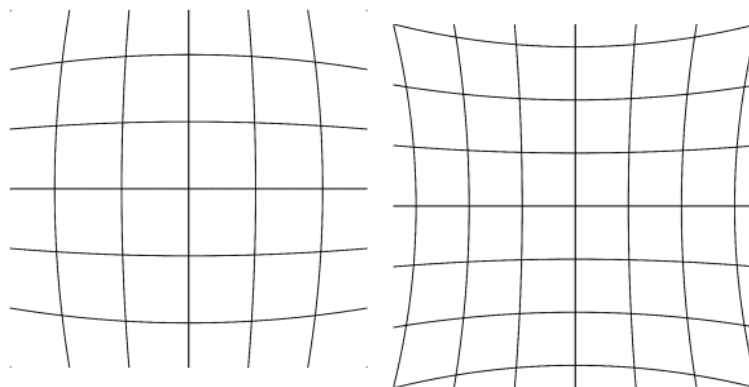


Figura 2.1: Cele mai întâlnite tipuri de distorsiuni (barrel – stânga și pincushion – dreapta) [1]

Date pe care le vom cunoaște legate de imagine:

- Distanța focală – distanța dintre centrul optic și planul imagine ( $f_x$ ,  $f_y$ )
- Punctul principal – coordonatele centrului real al imaginii, intersecția dintre axa optică și planul imagine (coordoanate date în pixeli) ( $u_0$ ,  $v_0$ )
- Coeficienți de distorsiune
  - Radială  $k_1$ ,  $k_2$

- Tangențială **p1, p2**

Principiul care stă la bază este corespondența unui pixel din imagine sursă la o altă locație în imaginea destinație

$$(x', y') = (x + \delta x, y + \delta y)$$

*Algoritmul de corecție*

Pentru fiecare pixel  $(u, v)$  din imaginea destinație D

- Se calculează  $(x, y)$  în planul imagine
  - $x = (u - u_0) / f_x$
  - $y = (v - v_0) / f_y$
- Se calculează coordonatele în imaginea distorsionată S:  $(x', y') = (x + \delta x, y + \delta y)$
- Se calculează coordonatele în pixeli în imaginea distorsionată S:
  - $u' = u_0 + x' f_x$
  - $v' = v_0 + y' f_y$
- Se atribuie pixelului destinație valoarea pixelului sursă în poziția găsită
 
$$D(u, v) = S(u', v')$$

Soluția naivă, cea mai simplificată ar fi maparea forward, presupunem că matricea A este sursa și B destinația [2]

$$B(u(x, y), v(x, y)) = A(x, y)$$

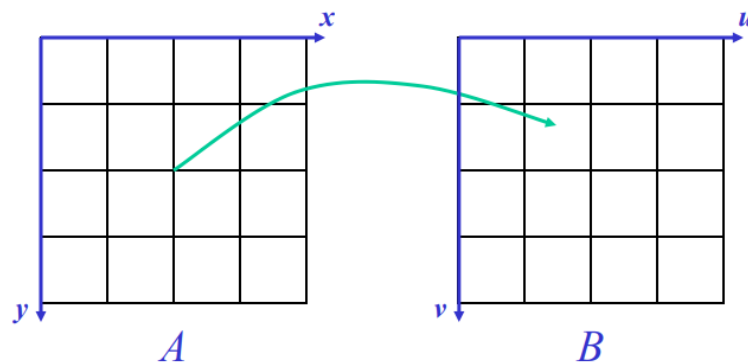


Figura 2.1: Maparea unui pixel din matricea A în matricea B [2]

Problema principală a acestei abordări este faptul că nu întotdeauna rezultatul va fi exact locația unui pixel deoarece rezultatul va fi un număr real, iar pixelii au coordonate numere întregi, așadar este nevoie de o aproximare. Această aproximare va cauza o pierdere a calității și va rezulta o imagine destinație pixelată. Soluția propusă pentru aceasta este să ”împrăștiem” efectul aplicat pe un anumit pixel și pe vecinătatea acestuia prin interpolare bilineară, după cum se poate observa în figura 2.2.

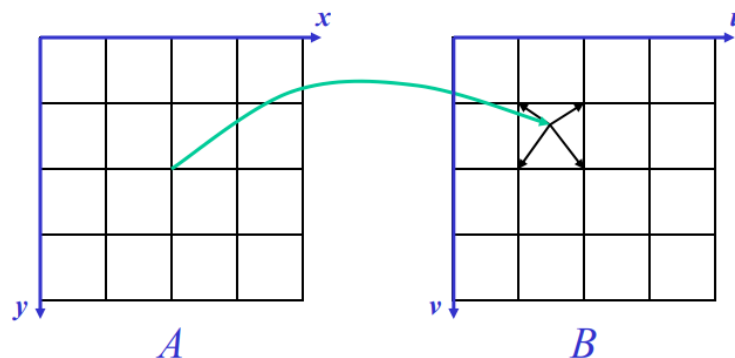


Figura 2.2: Soluția pentru problema interpolării forward cu interpolare bilineară

#### *Interpolarea bilineară – perspectiva matematică [3]*

Sau interpolarea 2-D este definită ca o interpolare liniară pe 2 axe ( $x$  și  $y$ ).

Să presupunem că avem punctele definite prin coordonatele  $(x_k, y_k)$  unde  $k = 1, 2$ .

Aceste puncte sunt locațiile lui  $Q_{11}$ ,  $Q_{12}$ ,  $Q_{21}$ ,  $Q_{22}$ . Pentru orice  $x$  și  $y$  dat care sunt între  $x_k$  și  $y_k$ , prin aplicarea interpolării bilineare putem găsi punctul  $P$  (definit de  $x$  și  $y$ ).

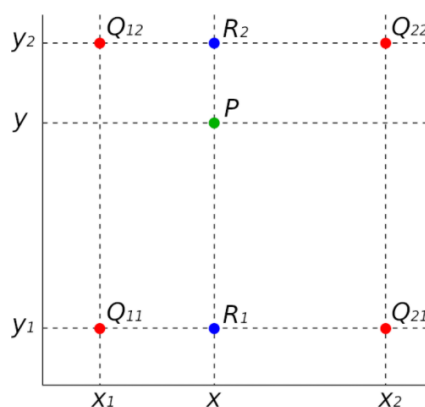


Figura 2.3: Principiul de funcționare a interpolării bilineare

Pentru a găsi punctul  $P(x, y)$  este nevoie de 2 etape

- 2 interpolări liniare pe axa  $x$ , pentru a găsi punctele  $R_1$  și  $R_2$  imediate
- 1 interpolare liniară pe axa  $y$  pentru a găsi punctul  $P$

Așadar, interpolarea bilineară constă în 2 interpolări, una pe axa  $x$  și alta pe axa  $y$ .

Punctul  $R1(x, y)$  va fi definit ca

$$R1(x, y) = Q11(x2-x) / (x2-x1) + Q21(x-x1) / (x2-x1)$$

$R2(x, y)$  va fi definit ca:

$$R2(x, y) = Q12(x2-x) / (x2-x1) + Q22(x-x1) / (x2-x1)$$

Punctul interpolat  $P(x, y)$  este definit ca:

$$P(x, y) = R1(y2-y) / (y2-y1) + R2(y-y1) / (y2-y1)$$

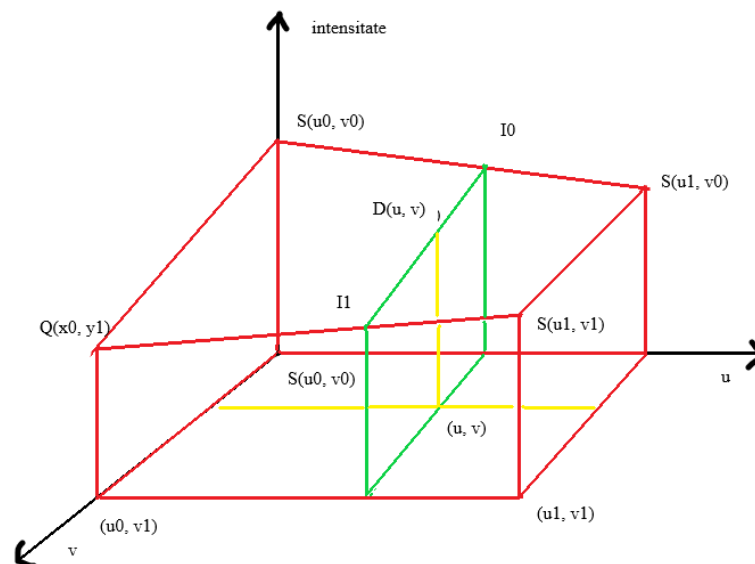


Figura 2.4: Exemplu vizual cu interpolarea biliniară a unui pixel

Imaginea de mai sus transpusă în pseudocod:

```
u0 = integer(u')
v0 = integer(v')
u1 = u0 + 1
v1 = v0 + 1
I0 = S(u0, v0) (u1 - u') + S(u0, v1) (u' - u0)
I1 = S(u0, v1) (u1 - u') + S(u1, v1) (u' - u0)
D(u, v) = I0 (v1 - v') + I1 (v' - v0)
```

### 3. Specificații de implementare

Parametrii intrinseci ai acestora sunt cunoscuți, ne oferă informații legate de distorsiunea radială (datorată curburii lentilelor –  $k_1$  și  $k_2$ ), tangențială (datorată nealinierii lentilelor –  $p_1$  și  $p_2$ ).

### 4. Rezultate experimentale

### 5. Concluzii

### 6. Bibliografie

- [1] H. Wolfgang, Correcting lens distortions in digital photographs, 2010.  
[http://www.imagemagick.org/Usage/lens/correcting\\_lens\\_distortions.pdf](http://www.imagemagick.org/Usage/lens/correcting_lens_distortions.pdf)
- [2] W. Wriggers, Interpolation and Morphing, The University of Texas.  
<http://www.biomachina.org/courses/imageproc/051.pdf>
- [3] Bilinear Interpolation  
<https://x-engineer.org/bilinear-interpolation/>
- [4] R. Dănescu, Modelul camerei. Procesul de formare a imaginilor  
[https://users.utcluj.ro/~rdanescu/pi\\_c02.pdf](https://users.utcluj.ro/~rdanescu/pi_c02.pdf)
- [5]