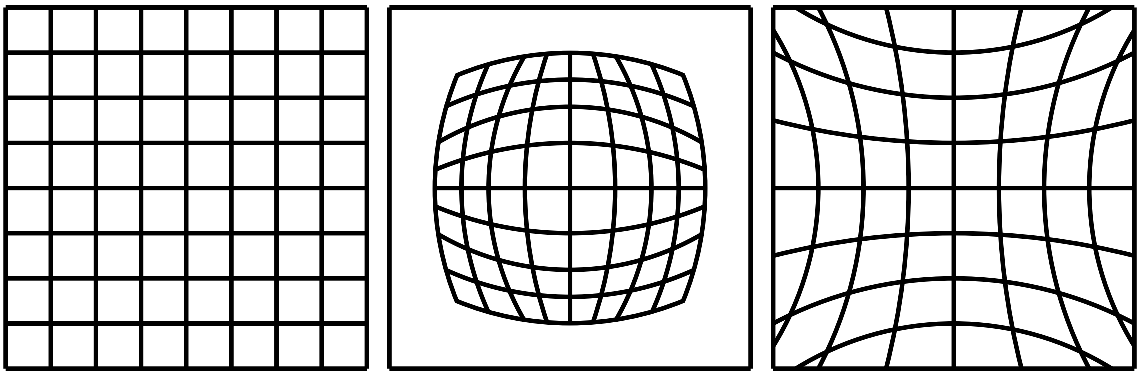
Odstranění geometrických zkreslení obrazu

Vstupní obraz pro naše úlohy získáváme pomocí optické soustavy tvořené objektivem a kamerou. Dle optických parametrů objektivu mohou v získaném obraze nastat geometrická zkreslení způsobená jeho optikou. Jedním z nejznámějších zkreslení je zkreslení typu *fisheye*, které vzniká u širokoúhlých objektivů s extrémně malou ohniskovou vzdáleností. Dalšími typy zkreslení jsou barel a polštář. Tato zkreslení jsou, spolu s nezkresleným obrazem, znázorněna na obrázku 1



Obrázek 1: Nezkreslený obraz, zkreslení typu barel (barrel) a polštář (pincushion)

Zkreslení typu barel je způsobeno vlastností širokoúhlých objektivů, která zapříčiňuje větší zvětšení středu obrazu než jeho okrajových částí. Podle výsledného efektu je také odvozen název deformace. Polštářový efekt je deformací inverzní.

Pokud bychom obraz, ve kterém je nějaký typ zkreslení, podrobili například detekci pohybu v obraze, výsledná detekce by trpěla chybou, kdy různé části takovéhoto obrazu by pro objekt pohybující se konstantní rychlostí vykazovaly rychlost objektu proměnlivou a to v závislosti na míře zkreslení.

Z takovýchto důvodů je žádoucí, aby bylo na zkresleném obrazu před aplikováním nějaké operace nejprve provedeno odstranění tohoto geometrického zkreslení.

V následující části popíšeme princip odstranění geometrického zkreslení, možnost jeho ručního nastavení a nakonec metodu, která automaticky najde nezkreslený obraz z původního deformovaného obrazu.

0.1 Popis metody odstranění zkreslení

Zkreslení typu barel nebo polštář jsou tzv. *radiální* zkreslení. Nejjednodušším způsobem jak namodelovat takovéto zkreslení je pomocí posunu souřadnic obrazu. Nechť $\bar{q} = (x_n, y_n)^T$ reprezentuje nezkreslené souřadnice, zatímco $q = (x_d, y_d)^T$ reprezentuje souřadnice pozorované (tedy souřadnice v deformovaném obraze). Každé radiálně symetrické zkreslení může být aproximováno pomocí Taylorova polynomu ve tvaru:

$$\phi(r^2) = 1 + K1r^2 + K2r^4 + \ldots, \tag{1}$$

kde $r^2 = \bar{x}^2 + \bar{y}^2$ a $K1$, $K2$, ... jsou koeficienty radiálního zkreslení.

Aby byly koeficienty nezávislé na velikosti vstupního obrazu, jsou souřadnice $\bar{x}$, $\bar{y}$ bezrozměrné. Navíc pro zachování radiální symetrie musíme přesunout počátek souřadnicového systému do středu obrazu. Souřadnice požadovaných vlastností obdržíme snadno s pomocí následujícího předpisu:

$$\begin{aligned} \bar{x} &= \tfrac{x_n - c_u}{R} \\ \bar{y} &= \tfrac{y_n - c_v}{R}, \end{aligned} \tag{2}$$

kde $R = \sqrt{c_u^2 + c_v^2}$ a souřadnice středu obrazu $c_u = w/2$, $c_v = h/2$ (w je šířka a h výška obrazu).

Výslednou transformaci přechodu ze souřadnic x_n, y_n rekonstruovaného obrazu do souřadnic x_d, y_d původního deformovaného obrazu lze zapsat následující rovnicí:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = \begin{pmatrix} x_n - c_u \\ y_n - c_v \end{pmatrix} \phi^{-1}(r^2) + \begin{pmatrix} c_u \\ c_v \end{pmatrix}. \tag{3}$$

Postupně procházíme výstupní obraz po jednotlivých pixelech $(x_n, y_n)^T$ a vypočítáváme souřadnice korespondujícího pixelu v nám známém vstupním deformovaném obraze $(x_d, y_d)^T$. Získané souřadnice jsou obecně reálná čísla, takže musíme zvolit vhodnou interpolaci. Jednoduchou interpolaci nejbližším sousedem docílíme přijatelného výsledku. Pro kvalitnější výstup je nutné využít interpolaci bilineární.

Výsledek dosažený pomocí popsané metody je zobrazen na obr. 2.

0.2 Popis metody aplikace zkreslení

V praxi exitují úholy, kde chceme do obrazu, který obsahuje nějaké geometrické zkreslení, vložit nové objekty. Takovýto postup je běžně využíván v počítačové grafice, kdy jsou do nasnímaného filmu vkládány nové objekty, např. vizuálních efektů, virtuálních postav, apod. Ve své podstatě jde tedy o postup opačný k dříve popsané metodě. Důvodem takovéhoto postupu je to, že jednou nasnímaný (kino)film má jednou provždy dané rozlišení. Počítačem generovaný obsah však můžeme renderovat i ve velmi vysokém rozlišení a takovýto obraz pak vkládat do původního (geometricky zkresleného) obrazu. Tímto způsobem je kvalita vstupního (kino)filmu zachována.

Obrázek 2: Původní zkreslený obraz (vlevo), obraz po odstranění radiálního zkreslení (vpravo)

K aplikaci zkreslení je nutné najít inverzní mapování funkce g^{-1} z rovnice 3. Řešení takovéto transformace však není možné najít analyticky a proto je nutné využít numerického řešení. V zásadě je cílem najít zkreslené souřadnice $(x_d, y_d)^T$ k nezkresleným souřadnicím $(x_n, y_n)^T$. Dále víme, že

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = g \begin{pmatrix} x_d \\ y_d \end{pmatrix}, \tag{4}$$

kde $g = \phi(r^2)$.

Rovnice iteračního algoritmu podle metody Newton-Raphson bude vypadat následovně:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix}^{(j+1)} = \begin{pmatrix} x_d \\ y_d \end{pmatrix}^{(j)} - \left(\frac{\partial g \left((x_d, y_d)^{(j)} \right)}{\partial \left((x_d, y_d)^{(j)} \right)} \right)^{-1} \left(g \begin{pmatrix} x_d \\ y_d \end{pmatrix}^{(j)} - \begin{pmatrix} x_n \\ y_n \end{pmatrix} \right). \tag{5}$$

Kvalita výsledku je závislá na dvou faktorech: počtu iterací a počáteční inicializaci. K dosažení dobrého výsledku zpravidla postačí 5 iterací, zatímco k dosažení přesného výsledku může být potřeba i 30 iterací. Jako počáteční podmínku $(x_d, y_d)^T$ je s úspěchem možno použít hodnotu $(0, 0)^T$. Tento iterační algoritmus je však nutno provádět pro každý pixel a proto je tento postup časově náročný.

0.3 Automatická kalibrace

Kdybychom chtěli nastavovat koeficienty radiálního zkreslení ručně, jistě bychom záhy přišli na to, že je tato operace poměrně náročná na přesnost a čas uživatele, který ji provádí. Existuje několik metod, které si kladou za cíl automatickou kalibraci zmíněných parametrů. Většina takových metod vyžaduje buď obrazy kalibrační mřížky nebo referenčního objektu, u nějž jsou známe rozměry. V praxi jsou však takovéto obrazy nedostupné a je třeba postupovat jiným způsobem.

V následujících odstavcích si popíšeme metodu, která vychází z jednoduchého předpokladu a poskytuje kvalitní výsledky.

Základním principem je tvrzení, že přímé linie ve snímané scéně by měly být přímé i ve výsledném (nezkresleném) obraze. Algoritmus, který využívá tohoto tvrzení, se též anglicky nazývá *plumb line*. Jednotlivé kroky algoritmu narovnávání je možno popsat následovně:

1. Uživatel ve zkresleném obraze vyznačí několik bodů tvořící čáru, která ve scéně tvoří přímku,

2. algoritmus hledá takové koeficienty radiálního zkreslení, při kterých jsou body co možná nejvíce v přímce,

3. pro nalezené koeficienty se aplikuje postup, kdy se každý bod zkresleného obrazu promítá do obrazu bez zkreslení.

0.3.1 Hledání koeficientů radiálního zkreslení

Hledání koeficientů radiálního zkreslení je opět prováděno iteračním algoritmem. Pseudokód algoritmu je popsán níže:

```
step = 0.1
threshold = 1.0e-4

while !good:
    error_minus = test( param - step )
    error_here = test( param )
    error_plus = test( param + step )

    if error_here is smallest:
        step /= 2
    if error_minus is smallest:
        param -= step
    if error_plus is smallest:
        param += step
    if step < threshold:
        good = True
```

Funkce `test` transformuje uživatelem definované body podle aktuální hodnoty koeficientu. Podle velikosti chyby, která je vrácena funkcí `test`, pak algoritmus upravuje parametr tak, aby ve výsledku postupně narovnával uživatelem definovanou lomenou čáru do přímky. Funkce `test` po transformaci bodů jimi proloží přímku pomocí metody nejmenších čtverců, která je popsána dále. Následně funkce vrací sumu vzdáleností vnitřních bodů definované lomené čáry od proložené přímky. Tato chyba je definována následujícím vzorcem:

$$\mathrm{Err} = \sum_i \sqrt{d_i^2}, \tag{6}$$

kde d_i je vzdálenost bodu od přímky.

Proložení přímky množinou bodů pomocí metody nejmenších čtverců

Mějme množinu n bodu (x_i, y_i), $(i = 1, 2, \ldots, n)$. Chceme najít takovou přímku $y = ax+b$, která nejlépe prokládá zadanou množinu bodů. Musíme proto najít parametry a, b, které nejlépe řeší předeterminovanou soustavu rovnic.

$$\begin{pmatrix} b_1 & x_1a_1 \\ b_2 & x_2a_2 \\ \vdots & \vdots \\ b_n & x_na_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{7}$$

Pro jednoduchost si popišme, jak můžeme koeficienty odpovídající zadání nalézt pomocí následujících dvou rovnic:

$$a = \frac{n\left(\sum xy\right) - \left(\sum y\right)\left(\sum x\right)}{n\left(\sum x^2\right) - \left(\sum x\right)^2}, \tag{8}$$

$$b = \left(\frac{1}{n}\sum y\right) - a\left(\frac{1}{n}\sum x\right) = \frac{\left(\sum y\right)\left(\sum x^2\right) - \left(\sum x\right)\left(\sum xy\right)}{n\left(\sum x^2\right) - \left(\sum x\right)^2}. \tag{9}$$

Pomocí takto získaných koeficientů a, b si vytvoříme přímku, se kterou budeme dále pracovat.

0.3.2 Odstranění zkreslení pomocí koeficientů získaných autokalibrací

V této chvíli již známe koeficienty radiálního zkreslení. Je nutné si uvědomit, že takto získané koeficienty nelze použít pro transformaci uvedenou v odstavci 0.1. Můžeme však provést transformaci opačnou, tedy ze souřadnic $(x_d, y_d)^T$ zkresleného obrazu do souřadnic $(y_n, y_n)^T$ obrazu nedeformovaného. K takové transformaci použijeme následující vztah:

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} c_u \\ c_v \end{pmatrix} + \begin{pmatrix} x_d - c_u \\ y_d - c_v \end{pmatrix} \phi(r^2). \tag{10}$$

Opět postupně procházíme vstupní obraz po jednotlivých pixelech $(x_d, y_d)^T$ a vypočítáváme souřadnice korespondujícího pixelu v nám neznámém výstupním nedeformovaném obraze $(x_n, y_n)^T$. Takto získaný výstup však trpí tím, že některým pixelům ve výstupním obraze nebude přiřazena funkční hodnota ze vstupního obrazu. Tento jev můžeme odstranit tak,

Obrázek 3: Původní zkreslený obraz s uživatelem definovanou lomenou čárou (vlevo), obraz po odstranění zkreslení pomocí automatické kalibrace (vpravo)

že buď použijeme metodu popsanou v odstavci 0.2 nebo můžeme využít jednoduchého triku v podobě supersamplingu. V takovém případě budeme vstupní deformovaný obraz procházet ne po celých pixelech, ale po jejich částech (subpixelech). V takovém případě je však nutné vstupní barevnou hodnotu opět interpolovat pomocí bilineární interpolace.

Výsledek popsané metody s nakreslenou kalibrační lomenou čárou je zobrazen na obr. 3.

0.4 Zadání úlohy

1. Implementujte problém korekce radiálního zkreslení metodou popsanou v odstavci 0.1. Koeficienty upravujte ručně dokud nedocílíte výsledku zobrazeného v příkladu.

2. Nadefinujte několik bodů v původním zkresleném obraze. Naimplementujte autokalibrační metodu popsanou v odstavci 0.3. Výstupní obraz s odstraněným zkreslením vytvořte pomocí supersamplingu ze vstupního obrazu.

3. Autokalibrační metodu doplňte tak, aby využívala iterační metodu aplikace zkreslení z odstavce 0.2.