GEOMETRIJSKE OPERACIJE

Geometrijske operacije mijenjaju prostorne odnose elemenata na slici. Ove operacije se mogu posmatrati kao kretanje objekata unutar slike. Za izvođenje geometrijskih operacija neophodne su dvije grupe algoritama: jedni koji definišu prostorne transformacije same po sebi, tj. zadaju način kretanja svakog piksela, i drugi koji omogućavaju gray-level interpolaciju.

Definisanje kretanja svakog piksela slike ponaosob je neracionalno, te je uobičajeno da se kretanje piksela specificira matematički, prostornom relacijom koja povezuje piksele izlazne sa pikselima ulazne slike. Opšte forma geometrijskih operacija je prema tome

$$b(x,y) = a(x',y') = a[f(x,y), g(x,y)]$$

gdje je $a(x,y)$ ulazna slika, $b(x,y)$ izlazna slika, a funkcije $f(x,y)$ i $g(x,y)$ specificiraju prostornu transformaciju.

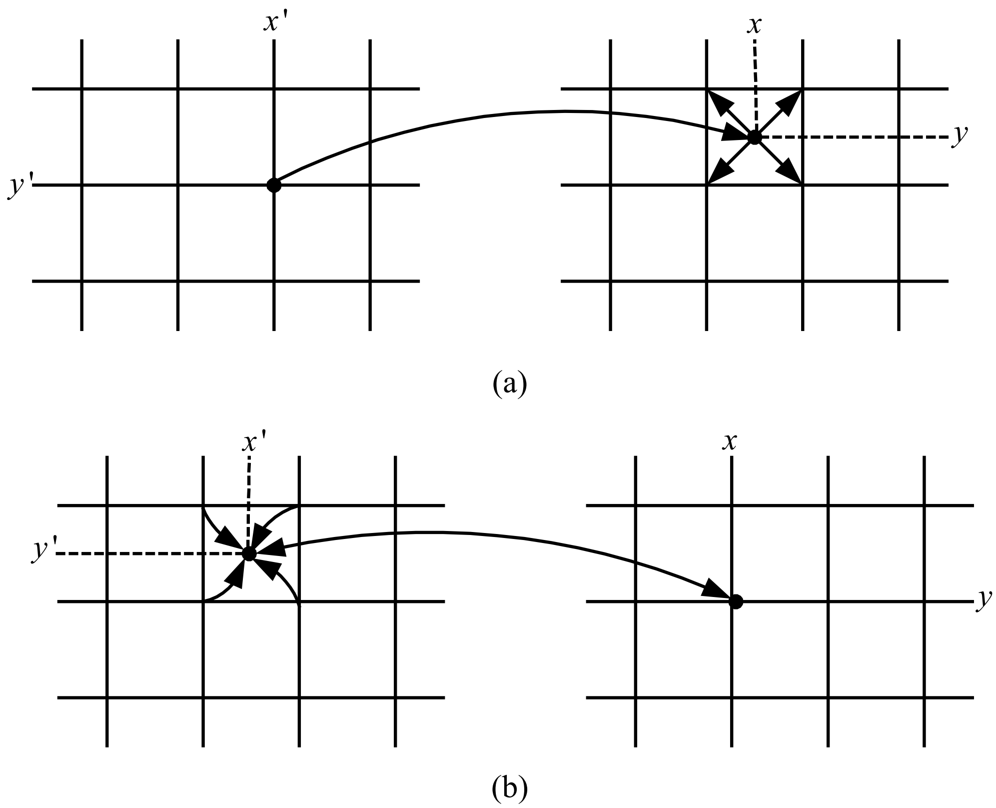
Iako su vrijednosti ulazne slike definisane samo u cjelobrojnim vrijednostima x i y, navedena jednačina može generisati vrijednosti izlazne slike na koordinatama koje nisu cijeli brojevi. Ako se neka geometrijska operacija posmatra kao mapiranje slike a u sliku b, pikseli slike a se mogu mapirati na pozicije između piksela na slici b i obrnuto. Mi ćemo (radi lakše diskusije) podrazumijevati da su pikseli locirani tačno na cjelobrojnim koordinatama.

Interpolacija nivoa sivila

Prilikom implementacije geometrijskih operacija može se smatrati da se vrši prenošenje nivoa sivila sa ulazne na izlaznu sliku, piksel po piksel. Ako se ulazni piksel mapira na poziciju između četiri izlazna piksela, njegov nivo sivila se dijeli između ta četiri piksela. Taj postupak nazivamo *mapiranje unapred*. Mnogo efikasnije je implementaciju posmatrati kao popunjavanje piksela izlazne slike ili *mapiranje unazad*. U ovom slučaju vrši se mapiranje izlaznih piksela u ulaznu sliku (jedan piksel u jednom trenutku) radi određivanja njihovih nivoa sivila. Ova dva načina generisanja nivoa sivila u pikselima izlazne slike su prikazana na Slici 78.

Algoritam mapiranja unapred je prilično neefikasan jer se dešava da se mnogi ulazni pikseli mapiraju van granica slike. Osim toga, svaki izlazni piksel se adresira više puta da bi se uzelo u obzir da njegov nivo sivila zavisi od više ulaznih piksela. Ako prostorna transformacija uključuje umanjenje slike, više od četiri ulazna piksela utiču na nivo sivila izlazne slike. Ako se radi uvećanje, neki izlazni pikseli mogu biti ispušteni jer ne postoje ulazni pikseli koji se mapiraju u njihovu blizinu.

Algoritam mapiranja unazad generiše izlaznu sliku piksel po piksel. Kako se izlazni piksel mapira najčešće u prostor između četiri ulazna piksela, neophodno je izvršiti interpolaciju da bi se odredio nivo sivila izlaznog piksela.



Slika 78. Mapiranje piksela: (a) unapred, (b) unazad

Interpolacija nultog reda

Najjednostavnija interpolacija se zove interpolacija nultog reda ili interpolacija na osnovu najbližeg susjeda. U ovom slučaju, nivo sivila izlaznog piksela se izjednači sa nivoom sivila onog ulaznog piksela koji je najbliži lokaciji gdje se mapira izlazni piksel. Algoritam je jednostavan i prihvatljiv u mnogim sitaucijama. Ipak, ovakva interpolacija može uvesti artifacte na slikama koje sadrže fine strukture i gdje se nivo sivila znatno mijenja od piksela do piksela.

Bilinearna interpolacija

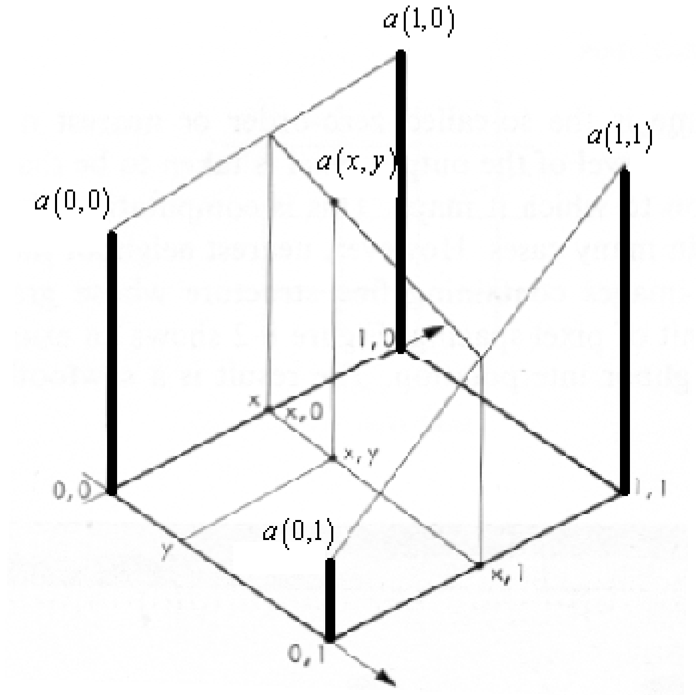
Interpolacija prvog reda, ili bilinearna interpolacija, daje mnogo bolje rezultate nego interpolacija nultog reda uz malo povećanje složenosti programiranja i povećanje vremena izvršavanja. Budući da je fitovanje ravni kroz četiri tačke preodređen problem, interpolacija prvog reda na pravougaonoj mrežici zahtijeva bilinearnu funkciju.

Neka je $a(x, y)$ funkcija dvije varijable sa poznatim vrijednostima u uglovima jediničnog kvadrata. Pretpostavimo da interpolacijom želimo da odredimo vrijednost $a(x, y)$ u bilo kojoj tački unutar jediničnog kvadrata, Slika 79. To možemo uraditi fitujući hiperbolički paraboloid definisan jednačinom:

$$a(x, y) = \mathrm{a}x + \mathrm{b}y + \mathrm{c}xy + \mathrm{d}$$

kroz četiri poznate vrijednosti.

Četiri koeficijenta a, b, c, d se biraju tako da $a(x, y)$ fituje poznate vrijednost u uglovima jediničnog kvadrata.



Slika 79. Bilinearna interpolacija

Algoritam se svodi na sljedeće. Prvo izvršimo linearnu interpolaciju između tačaka $a(0,0)$ i $a(1,0)$ koja omogućava da se odrede vrijednosti na gornjoj lijevoj ivici jediničnog kvadrata:

$$a(x,0) = a(0,0) + [a(1,0) - a(0,0)]x .$$

Na sličan način, za vrijednosi na donjoj desnoj ivici jediničnog kvadrata dobijemo:

$$a(x,1) = a(0,1) + [a(1,1) - a(0,1)]x .$$

Konačno vršimo linearnu interpolaciju

$$a(x, y) = a(x,0) + [a(x,1) - a(x,0)]y .$$

Uvrštavajući $a(x,0)$ i $a(x,1)$ u poslednju jednačinu imamo:

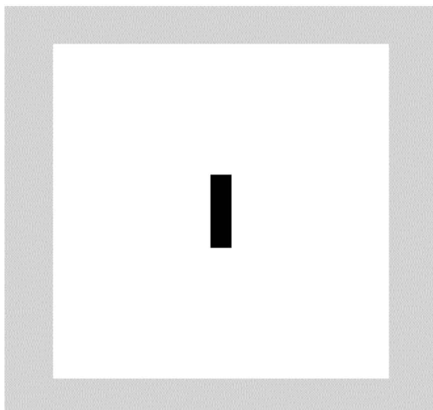
$$a(x, y) = [a(1,0) - a(0,0)]x + [a(0,1) - a(0,0)]y + [a(1,1) + a(0,0) - a(0,1) + a(1,0)]xy + a(0,0)$$

u formi bilinearne jednačine. Napomenimo da za *x* ili *y* konstantno, ova jednačina postaje linearna po drugoj varijabli, što znači da presijecanje hiperboličnog paraboloida nekom ravni koja je paralelna *xz*-ravni ili *yz*-ravni daje pravu.

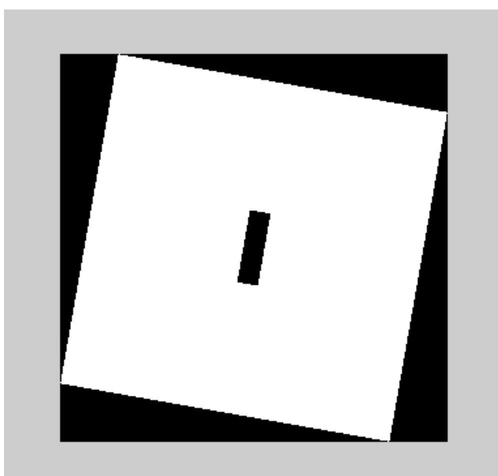
Slika 80 prikazuje interpolaciju nultog i prvog reda na primjeru rotacije. Vidljivo je da su ivice su više "nazubljene" pri interpolaciji nultog reda.

Interpolacije višeg reda

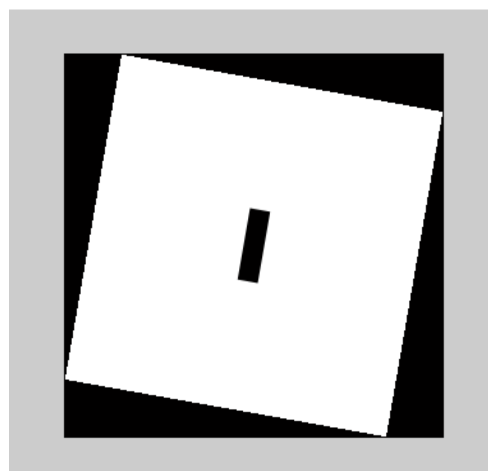
Pri geometrijskim operacijama, smoothing efekt bilinearne interpolacije nivoa sivila može degradirati fine detalje slike, posebno kada se radi uvećanje slike. Bolji rezultati se postižu interpolacijama višeg reda, funkcijama sa više od četiri koeficijenta koji zavise od više od četiri susjedna piksela. Ako je broj koeficijenata jednak broju tačaka, interpolirajuća funkcija prolazi kroz sve zadate tačke. U slučajevima kada je broj koeficijenata manji od broja tačaka, onda interpolirajuća funkcija prolazi što je moguće bliže zadatim tačkama, po metodu minimalne srednjekvadratne greške. Slika 80 prikazuje rotaciju.



(a)



(b)



(c)

Slika 80. Rotacija slike sa interpolacijom. (a) originalna slika, (b) rotacija sa interpolacijom nultog reda, (c) rotacija sa bilinearnom interpolacijom.

Prostorne transformacije

Jednostavne prostorne transformacije

Ako u opštu jednačinu

$$b(x,y) = a(x',y') = a[f(x,y), g(x,y)]$$

koja opisuje prostorne transformacije stavimo da je

$$f(x,y) = x \qquad g(x,y) = y$$

dobićemo *identitet*, jednostavno kopiranje slike a u sliku b.

U slučaju da je

$$f(x,y) = x + x_0 \qquad g(x,y) = y + y_0$$

radi se o *translaciji* pri kojoj se tačka x_0, y_0 translira u ishodište, a svi objekti unutar slike se pomijeraju za iznos $\sqrt{x_0^2 + y_0^2}$.

Koristeći formulaciju koja se naziva *homogene koordinate*, možemo razmatranja iz xy $(z = 0)$ ravni zamijeniti razmatranjima u $z = 1$ ravni trodimenzionalnog prostora i koristiti matričnu formu zapisa jednačina translacije u obliku:

$$\begin{bmatrix} f(x,y) \\ g(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Kada je

$$f(x,y) = \frac{x}{c} \qquad g(x,y) = \frac{y}{d}$$

skaliramo sliku faktorom c po x pravcu i faktorom d po y pravcu. Koordinatni početak slike (tipično gornji lijevi ugao) ostaje stacionaran pri *povećanju* ili *smanjenju* slike.

Zapisano u homogenim koordinatama skaliranje je dato sa:

$$\begin{bmatrix} f(x,y) \\ g(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{c} & 0 & 0 \\ 0 & \frac{1}{d} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Ako je $c = -1$ i $d = 1$ imamo *refleksiju oko y-ose*:

$$f(x, y) = -x \quad g(x, y) = y ,$$

a ako je $d = -1$ i $c = 1$ *refleksiju oko x-ose*:

$$f(x, y) = x \quad g(x, y) = -y .$$

Konačno, stavljajući da je:

$$f(x, y) = x\cos(\theta) - y\sin(\theta),$$
$$g(x, y) = x\sin(\theta) + y\cos(\theta),$$

dobivamo *rotaciju* za ugao θ oko ishodišta.

Zapisujući poslednje jednačine u homogenim koordinatama imamo:

$$\begin{bmatrix} f(x, y) \\ g(x, y) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Jasno, možemo kombinovati navedene osnovne geometrijske operacije, npr., translaciju sa uvećanjem da bi dobili sliku koja "raste" oko neke tačke koja nije ishodište. Slično, kombinacija translacije i rotacije daje rotaciju oko proizvoljne tačke.

Homogene koordinate omogućavaju da se odredi jedinstvena matrica za kombinovane operacije. Na primjer, rotacija oko tačke x_0, y_0 se može izvesti kao translacija tako da tačka x_0, y_0 padne u ishodište, zatim rotacija oko ishodišta za ugao θ i ponovna translacija na originalnu poziciju:

$$\begin{bmatrix} f(x, y) \\ g(x, y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

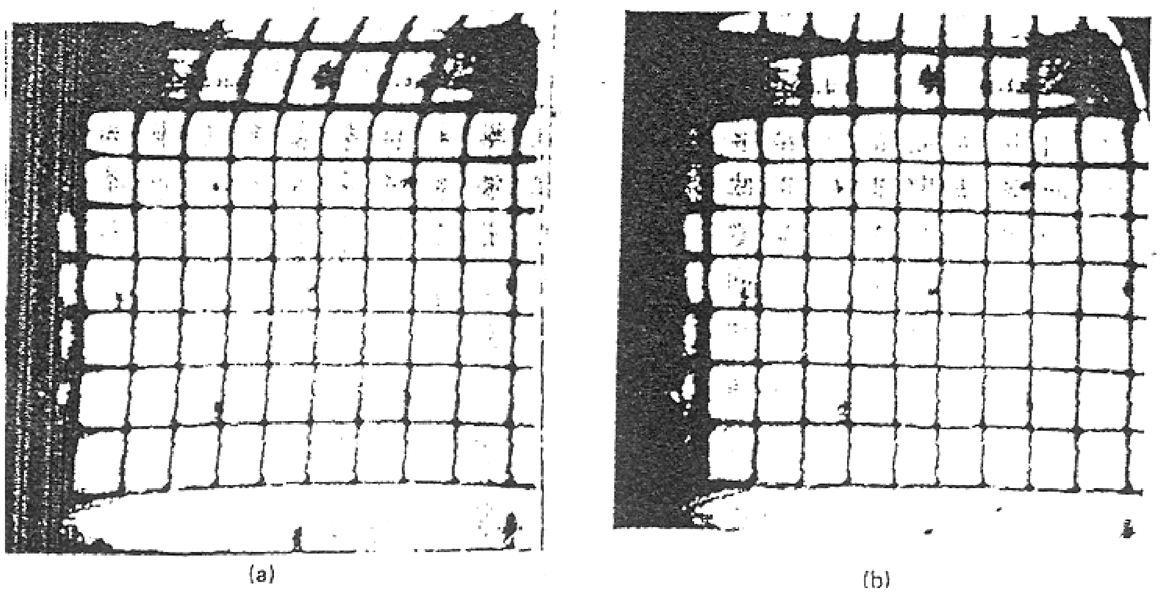
Na sličan način se formiraju i druge složene operacije.

Separabilna implementacija

Ako je slika podvrgnuta translaciji ili skaliranju, adrese izlaznih piksela $f(x, y)$ i $g(x, y)$ zavise samo od x i y, respektivno. To znači da je moguće, i ponekad mnogo efikasnije, te operacije podijeliti u dva koraka. Razmatrana operacija se prvo izvede u jednom, recimo horizontalnom, smjeru i dobije se međuslika, nad kojom se onda izvodi razmatrana operacija u ortogonalnom (vertikalnom) smjeru koja daje konačan rezultat.

Transformacije na osnovu kontrolnih tačaka

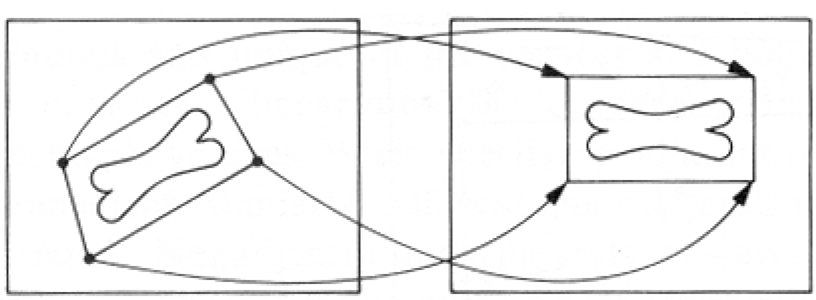
U mnogim aplikacijama obrade slike, prostorne transformacije su relativno složene i teško se mogu opisati matematičkim izrazima. Osim toga, željena translacija piksela se često određuje na osnovu mjerenja. Kao primjer možemo posmatrati geometrijsku kalibraciju slike koja je dobivena kamerom sa geometrijskom distorzijom. Slika kvadratne mrežice dobivena takvom kamerom prikazana je na Slici 81(a).



Slika 81. [3] Geometrijska kalibracija: (a) slika mrežice, (b) slika mrežice bez distorzije

Željena prostorna transformacija je ona koja će od slike kvadratne mrežice dobivene kamerom sa distorzijom generisati sliku kvadratne mrežice bez distorzije, Slika 81(b). Tako određena prostorna transformacija se onda koristi na svim slikama koje su dobivene istom kamerom.

Uobičajeno je da se ne specificira vrijednost pomjeraja svakog piksela, već da se to uradi na skupu kontrolnih tačaka, Slika 82.



Slika 82. [3] Mapiranje kontrolnih tačaka

Pomjeraj nespecificiranih tačaka se određuje interpolacijom. Najčešće se to radi bilinearnom transformacijom. Opšti oblik bilinearne prostorne transformacije je:

$$b(x,y) = a(x',y') = a(\mathrm{a}x + \mathrm{b}y + \mathrm{c}xy + \mathrm{d}, \mathrm{e}x + \mathrm{f}y + \mathrm{g}xy + \mathrm{h}).$$

Bilinearna transformacija je definisana sa 8 vrijednosti parametara a,b,c,d,e,f,g,h. Specificirajući u koje četiri tačke koje odgovaraju uglovima pravougaonika se mapiraju odabrane četiri tačke koje odgovaraju uglovima četverougla (sl. 49), kreiramo dva skupa od po četiri linearne jednačine sa četiri nepoznate. Mapiranje *x*' u *x* generiše četiri jednačine iz kojih se računaju koeficijent a,b,c i d, dok mapiranje *y*' u *y* generiše četiri jednačine iz kojih se računaju koeficijent e,f,g i h.

Postoji mnogo efikasniji način za implementaciju ovakvih prostornih transformacionih algoritama od onog datog prethodnom jednačinom. Data jednačina se može zapisati na sljedeći način:

$$b(x,y) = a(x + dx(x,y), y + dy(x,y)),$$

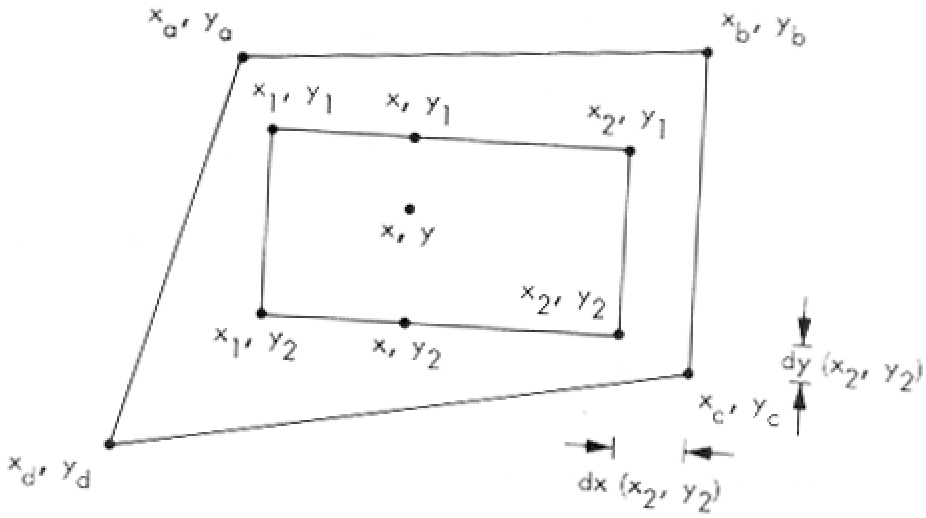
gdje su $dx(x,y)$ i $dy(x,y)$ pomjeraji koji su bilinearne funkcije od *x* i *y*. Slika 83 prikazuje ove pomjeraje. Problem se sada svodi na specificiranje *dx* i *dy* za sve tačke unutar pravougaonika. Kako su $dx(x,y)$ i $dy(x,y)$ bilinearni po *x* i *y*, oni postaju linearni duž svake horizontalne ili vertikalne linije na slici. Prema tome, za svaku horizontalnu liniju možemo definisati inkrement Δx takav da, podrazumijevajući jedinični razmak piksela, vrijedi:

$$dx(x+1,y) = dx(x,y) + \Delta x,$$

$$\Delta x_1 = \frac{dx(x_2,y_1) - dx(x_1,y_1)}{N},$$

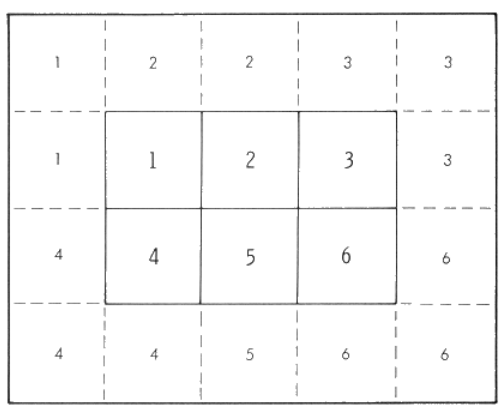
gdje je N broj piksela po *x* osi od tačke (x_1, y_1) do tačke (x_2, y_1).

Slično se uradi za *dy*. Inkrement Δx se razlikuje od linije do linije, ali se lako izračuna na osnovu vrijednosti pomjeraja na krajevima pravougaonika. Implementacija poslednje jednačine zahtijeva samo dva sabiranja, jedno za *dx* i jedno za *dy* da bi se za svaki izlazni piksel izračunale koordinate ulaznog piksela.



Slika 83. [3]   Pomijeranja kontrolnih tačaka

Navedena procedura daje prostornu transformaciju za tačke koje padaju unutar izlaznog pravougaonika. Često je ovo jednostavno mapiranje četverougla u pravougaonik neadekvatno da bi se specificirala željena prostorna transformacija. Tada je moguće generisati skup četverouglova u ulaznoj slici koji se mapiraju u skup pravougaonika izlazne slike. Nije neophodno da se pravougaonicima prekrije cijela slika. Slika 84 prikazuje izlaznu sliku na kojoj je specificirano šest pravougaonika. Unutar svakog od njih korištena je opisana prostorna transformacija na osnovu kontrolnih tačaka. Na slici je takođe prikazano kako se vrši ekstrapolacija prostornih transformacija van specificiranih pravougaonika. Broj unutar pravougaonika za koje nije specificirana prostorna transformacija određuje pravougaonik čiji bilinearni koeficijenti se koriste za prostornu transformaciju.



Slika 84. [3]   Ekstrapolacija