



ZOBRAZENIE A SPRACOVANIE DÁT V ROVINE

MICHAL LEKÝR : PETER TARÁBEK

Publikácia bola vydaná s finančnou podporou projektu VEGA: 1/0077/22 „Inovatívne predikčné metódy pre optimalizáciu verejných obslužných systémov“.

Autori: Ing. Michal Lekýr, PhD., Ing. Peter Tarábek, PhD.

Pracovisko autorov:

Katedra matematických metód a operačnej analýzy
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Univerzitná 1
010 26 Žilina

Za odbornú, jazykovú a technickú úroveň publikácie zodpovedá autor.
Vydala Žilinská univerzita v Žiline / EDIS – vydavateľstvo UNIZA

© Michal Lekýr, Peter Tarábek, 2024

Obsah

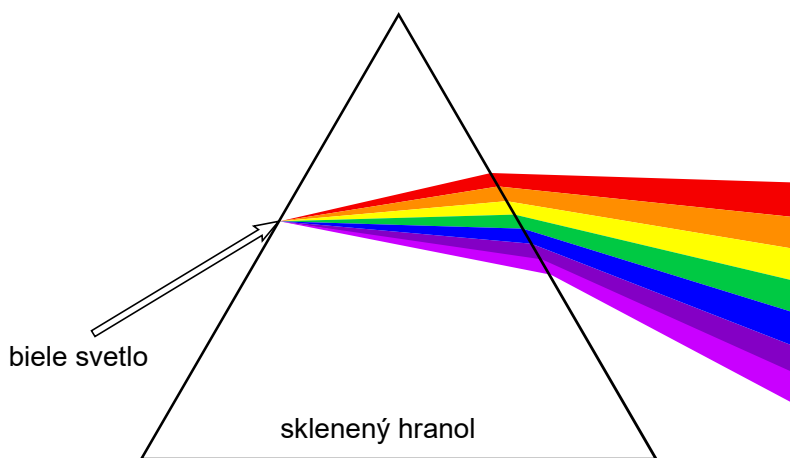
1. Vnímanie svetla a farieb	6
1.1. Ľudský vizuálny systém	11
1.2. Vlastnosti farieb	17
2. Základy digitálneho obrazu	20
2.1. Rastrová a vektorová reprezentácia digitálneho obrazu	21
2.2. Matematický model obrazu a digitalizácia	28
2.3. Rozlíšenie digitálneho obrazu	29
2.4. Základné vzťahy medzi pixelmi	34
2.5. Farebné modely	37
3. Základné metódy spracovania obrazu	42
3.1. Konvolúcia a filtrovanie obrazu	43
3.2. Redukcia šumu a vylepšenie obrazu	50
3.3. Prahovanie a detekcia hrán	54
3.4. Morfológické spracovanie obrazu	61
4. Antialiasing a rasterizácia	66
4.1. Miestny alias	72
4.2. Časový alias	74
4.3. Antialiasing bodov a čiar	76
4.4. Rasterizácia DDA	80
4.5. Bresenhamov algoritmus	83
4.6. Rasterizácia čiar pomocou Xiaolin Wu algoritmu	86
5. Grafické dáta.....	87
5.1. Rovinné grafické dáta	91
5.2. Ukladanie a spracovanie grafických dát	94
5.3. Lineárna algebra pre prácu s dátami	98
5.4. Vektory	102
5.5. Matice	105
5.6. Výpočtové operácie	109

6.	Transformácie na ploche	
6.1.	Súradnicové systémy	112
6.2.	Posunutie	114
6.3.	Otočenie	116
6.4.	Zmena mierky	119
6.5.	Skosenie a zrkadlenie	121
6.6.	Optimalizácia výpočtového výkonu	124
7.	Krivky	
7.1.	Explicitné, implicitné, parametrické	125
7.2.	Spojitosť kriviek	127
7.3.	Hermitovské kubické krivky	131
7.4.	Beziérová krivka	135
7.5.	Splajny	138
7.6.	Využitie kriviek na animáciu objektov	142
8.	Záver.....	145
8.1.	Prílohy	147
8.2.	Literatúra	155

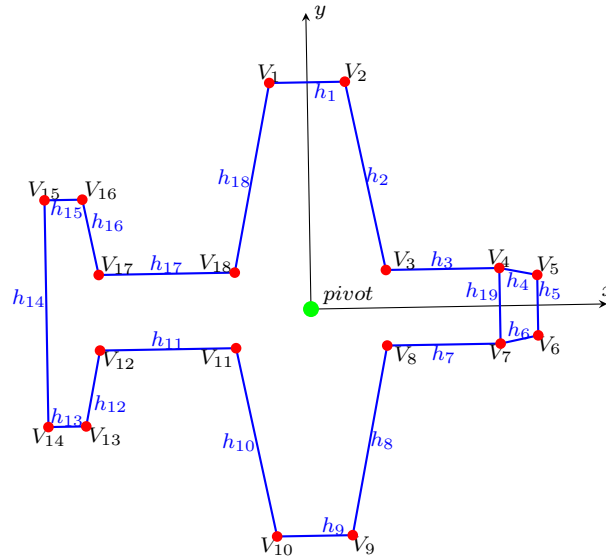
Vnímanie svetla a farieb

1. Ľudský vizuálny systém

Pojem farba je na prvý pohľad veľmi jasný a intuitívny. Môžeme ukázať na stôl a povedať, že stôl je hnedý. Základné farby si vieme jednoducho predstaviť. Napríklad povieme zelená farba, čitateľ má pomerne jasnú predstavu, čo tým myslíme. V skutočnosti je však odpoveď na otázku „Čo je farba?“ oveľa zložitejšia. Farba je psychofyziologický vnem vyvolaný vstupom viditeľného svetla určitej vlnovej dĺžky do oka. Už z tejto stručnej definície je zrejmé, že v prípade farby je potrebné sa zaoberať nie len jej fyzikálnou ale aj psychologickou podstatou. O fyzikálnych vlastnostiach svetla máme pomerne dobré znalosti. Mechanizmy, ktorými svetlo reaguje s orgánmi zraku, nie sú až tak dobre pochopené. Otázka farby je veľmi stará. Napríklad už Newton vedel, že svetelné lúče nie sú zafarbené. V 60. rokoch 16. storočia začal tento anglický fyzik a matematik sériu experimentov so slnečným svetlom a hranolmi. V tom čase si ľudia mysleli, že farba je zmes svetla a tmy a že hranoly zafarbujú svetlo. Newton si uvedomil, že táto teória bola nesprávna. Pri svojom okne postavil hranol a toto spektrum premietol na vzdialenú stenu (obr.: 1). Mimochodom podobný jav, dúhu, môžeme pozorovať pri lome svetla medzi kvapkami vody a vzduchu. Aby dokázal, že hranol svetlo nezafarbujú, použil ďalší hranol aby spektrum zložil späť do bieleho slnečného svetla. To znamená, že neexistuje entita svetlo, ku ktorej stačí pridať vlastnosť zelená a tým dostaneme zelené svetlo.



OBR. 1. Svetelný experiment Isaaca Newtona

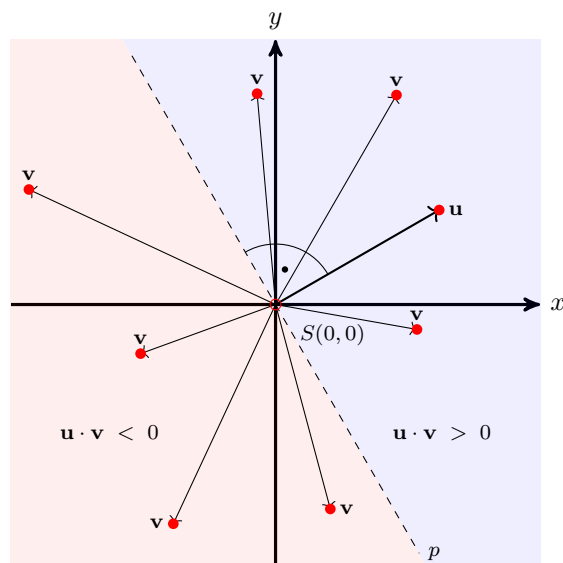


OBR. 1. Zobrazenie geometrických dát v podobe vrcholov a topologických dát v podobe hrán.

- **Index buffer (IBO alebo EBO)** – obsahuje indexy do poľa vrcholov, ktoré určujú poradie ich zoskupenia do geometrických entít (napr. body, čiary, trojuholníky, ale aj iné, komplexnejšie priestorové štruktúry).
- **Topologické zoznamy** – napríklad zoznamy susednosti, grafové reprezentácie objektov alebo dátové štruktúry typu "face-edge-vertex" známe z CAD systémov.

Ak sa teda vrátíme k predošlému obrázku zobrazujúcemu lietadlo, môžeme skonštruovať pamäťové miesta pre uloženie týchto údajov. Vyalokované sektory musia okrem iných dát obsahovať aj indexy prvkov. Nakoľko sa jedná o údaje pevného rozsahu, môžeme stĺpec s indexom vynechať a nahradiť ho výpočtom adresy konkrétneho prvku v tabuľke. Napríklad ak budú prvky adresované pomocou 32 bitovej adresy a veľkosť štruktúry bude 2 x reálne číslo v jednoduchej presnosti, potom adresu prvku získame vynásobením indexu prvku a pripočítaním konštanty 64. Tento istý princíp platí pre dáta nachádzajúce sa v tabuľke indexov.

V tomto jednoduchom príklade však uvádzame len minimálny rozsah uložených dát. V reálnych aplikáciach sa rozsah dát určuje podľa potreby a umožňuje ho tzv. flexibilný formát vertexov (flexible vertex format - FVF), ktorý popíšeme v ďalšej časti. Treba tiež uviesť, že okrem možnosti rozšíriť údaje vrcholov o ďalšie atribúty, samotná štruktúra indexovej tabuľky sa mení nielen na základe topológie objektu, ale aj na základe formy vykreslenia objektu. Na tomto príklade je ukázané, ako možno uložiť dáta pre vykreslenie tzv. drôteného modelu. Ak by sme lietadlo chceli vykresliť pomocou siete trojuholníkov, štruktúra indexovej tabuľky by sa



OBR. 6. Znamienko skalárneho súčinu nám hovorí, či dva vektory ležia v tej istej polrovine určenej osou p , alebo ležia každý v inej polrovine.

bod alebo smer, definícia nadobúda tvar

$$(6.7) \quad \|\mathbf{u}\| = \sqrt{u_x^2 + u_y^2 + u_z^2}$$

Vektor s dĺžkou presne 1 má *jednotkovú dĺžku* (je to *jednotkový vektor*). Každý vektor $\mathbf{u}$ s aspoň jednou nenulovou zložkou môžeme *znormalizovať* na jednotkovú dĺžku násobením faktorom $1/\|\mathbf{u}\|$:

$$(6.8) \quad \hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \quad \mathbf{u} \neq \mathbf{0}$$

Tento proces nazývame *normalizácia* a v počítačovej grafike sa používa veľmi často. Termín *normalizovať* nijako nesúvisí s pojmom *normála* (normálový vektor je kolmý na danú plochu v konkrétnom bode ako to vidíme na obr. ??). Normalizácia oddeľuje *smer* od *veľkosti*, čo zjednodušuje výpočty a zlepšuje numerickú stabilitu. Uveďme niekoľko príkladov z oblasti počítačovej grafiky, kde normalizácia vektorov zohráva kľúčovú rolu:

- **Výpočet uhla na základe skalárneho súčinu vektorov.** Pre jednotkové vektory $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ platí

$$(6.9) \quad \hat{\mathbf{u}} \cdot \hat{\mathbf{v}} = \cos \alpha$$

Počítanie uhla medzi vektormi je častá operácia, hlavne pri výpočte osvetlenia, tieňovania objektov a určovaním viditeľnosti (už spomínaný *backface culling*). Použitím normalizovaného vektora z pôvodného vzorca vypadne deliteľ a pri každom výpočte ušetríme jednu operáciu násobenia a jednu operáciu delenia. Normálové vektory sa štandardne počítajú dopredu pri

pričom zvolený stupeň regresného polynómu musí byť menší, ako počet zadanych bodov $k < n$.

Minimalizujeme kritérium

$$(4.5) \quad C = \min \sum_{i=1}^n (P(x_i) - y_i)^2$$

Úlohou je nájsť $k+1$ neznámych koeficientov regresného polynómu $a_0, a_1, a_2, \dots, a_k$ tak, aby hodnota kritéria C bola minimálna. C je funkcia $k+1$ premenných $a_0, a_1, a_2, \dots, a_k$ a pre minimálne hodnoty funkcie C teda musí platiť.

$$(4.6) \quad \frac{\partial C}{\partial a_i} = 0, \quad i = 0, 1, \dots, k.$$

Po dosadení za C a rozpísaní týchto podmienok dostávame sústavu rovníc

$$(4.7) \quad \begin{aligned} \frac{\partial}{\partial a_0} \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_k x_i^k - y_i)^2 &= 0 \\ \frac{\partial}{\partial a_1} \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_k x_i^k - y_i)^2 &= 0 \\ &\vdots \\ \frac{\partial}{\partial a_k} \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_k x_i^k - y_i)^2 &= 0 \end{aligned}$$

Po vykonaní derivácií dostávame sústavu lineárnych algebraických rovníc pre výpočet regresného polynómu $a_0, a_1, a_2, \dots, a_k$. Po minimalizácii súčtu štvorcov chýb a zderivovaní podľa jednotlivých koeficientov $a_0, a_1, \dots, a_k$ dostávame sústavu normálnych rovníc.

$$(4.8) \quad 2 \sum_{i=1}^n (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_k x_i^k - y_i) x_i^j = 0, \quad j = 0, 1, \dots, k$$

Po vydelení rovníc číslom 2 a úprave dostávame lineárnu sústavu:

$$(4.9) \quad \sum_{m=0}^k a_m \sum_{i=1}^n x_i^{m+j} = \sum_{i=1}^n y_i x_i^j, \quad j = 0, 1, \dots, k$$

Pre zjednodušenie zápisu zavedieme pomocné súčty:

$$(4.10) \quad S_j = \sum_{i=1}^n x_i^j, \quad j = 0, 1, \dots, 2k, \quad T_j = \sum_{i=1}^n y_i x_i^j, \quad j = 0, 1, \dots, k$$

Potom má sústava normálnych rovníc tvar:

$$(4.11) \quad \begin{aligned} a_0 S_0 + a_1 S_1 + \dots + a_k S_k &= T_0, \\ a_0 S_1 + a_1 S_2 + \dots + a_k S_{k+1} &= T_1, \\ &\vdots \\ a_0 S_k + a_1 S_{k+1} + \dots + a_k S_{2k} &= T_k \end{aligned}$$

$$\|\mathbf{H}'(\tfrac{1}{2})\| = \sqrt{1.55^2 + 0.4^2} = \sqrt{2.5625} \approx 1.6008,$$

samotný normalizovaný dotyčnicový vektor je

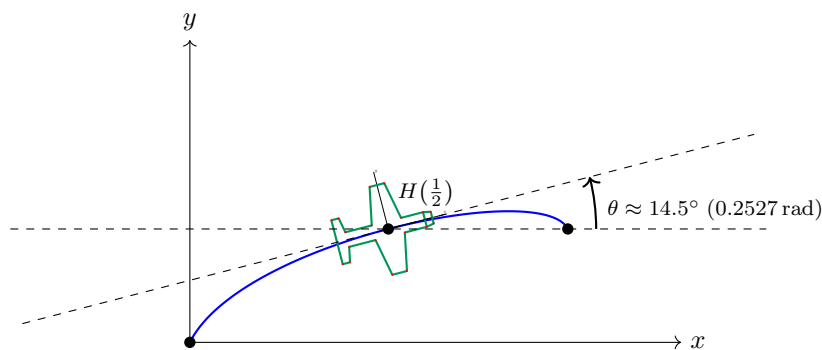
$$\mathbf{T}_{\frac{1}{2}} = \frac{1}{1.6008} (1.55, 0.4) \approx (0.9683, 0.2499).$$

Uhol orientácie objektu v smere pohybu vypočítame ako

$$\theta_{t=\frac{1}{2}} = \text{atan2}(0.4, 1.55) = \text{atan2}(0.2499, 0.9683) \approx 0.2527 \text{ rad} \approx 14.5^\circ.$$

Poznamenajme, že na samotný výpočet uhla orientácie objektu nie je potrebné normalizovať dotyčnicový vektor, keďže funkcia atan2 závisí iba od pomeru jeho zložiek. Normalizácia sa vykonáva z praktických dôvodov, aby sme získali jednotkový smerový vektor nezávislý od rýchlosti zmeny parametra. Takto je možné oddeliť smer pohybu od jeho rýchlosti a jednotkový vektor následne použiť na orientáciu objektu, výpočet normál alebo ďalšie geometrické operácie.

Aj keď normalizácia dotyčnicového vektora poskytuje jednotkový smer pohybu, nezaručuje konštantnú rýchlosť pohybu objektu po krivke. Pri konštantnom kroku parametra Δt sa rýchlosť pohybu mení v závislosti od veľkosti derivácie $\|\mathbf{H}'(t)\|$, ktorá je ovplyvnená tvarom a zakrivením krivky. Objekt sa preto na niektorých úsekoch krivky pohybuje rýchlejšie a na iných pomalšie. Na dosiahnutie konštantnej rýchlosti by bolo potrebné použiť parametrizáciu podľa dĺžky oblúka, ktorá však pre kubické polynomiálne krivky nemá uzavretý analytický tvar a v praxi sa realizuje numerickými aproximáciami.



OBR. 8. Pri animácii objektu po krivke potrebujeme okrem jeho polohy v čase $t = \frac{1}{2}$ vypočítať aj uhol otočenia objektu v danom čase. Vypočítaný uhol má hodnotu $\theta_{t=\frac{1}{2}} \approx 14.5^\circ$ (0.2527 rad).

V tejto sekcii sme sa pokúsili ukázať na príklade Hermitovskej krivky proces návrhu a následnej a parametrizácie kubickej krivky pomocou počiatočného bodu, koncového bodu a dotyčnicových vektorov. Tento princíp platí aj pre iné, používané typy kriviek, ktoré si popíšeme v nasledujúcich podkapitolách.

Správnosť výsledku môžeme overiť priamym použitím Bernsteinových polynómov. Pre kubickú krivku a $t = \frac{1}{2}$ majú jednotlivé váhy hodnoty

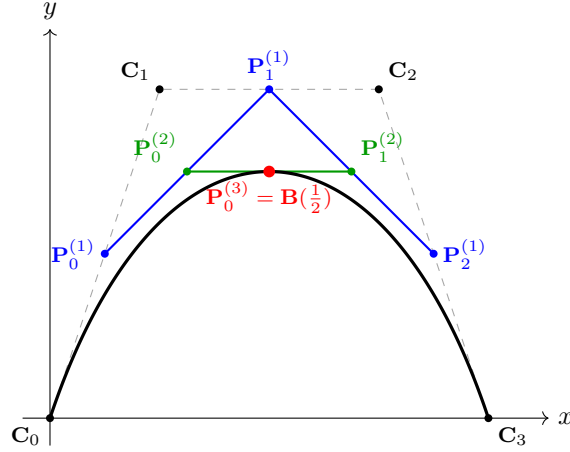
$$B_0^3(\frac{1}{2}) = \frac{1}{8}, \quad B_1^3(\frac{1}{2}) = \frac{3}{8}, \quad B_2^3(\frac{1}{2}) = \frac{3}{8}, \quad B_3^3(\frac{1}{2}) = \frac{1}{8}.$$

Po dosadení riadiacich bodov dostaneme

$$(7.22) \quad \begin{aligned} \mathbf{B}(\frac{1}{2}) &= \frac{1}{8}(0,0) + \frac{3}{8}(1,3) + \frac{3}{8}(3,3) + \frac{1}{8}(4,0) \\ &= \left(2, \frac{9}{4}\right), \end{aligned}$$

čo je rovnaký výsledok ako pri použití algoritmu de Casteljau.

Jednotlivé úrovne geometrickej konštrukcie sú znázornené na obrázku ?? . Sivou prerušovanou čiarou je zobrazený pôvodný riadiaci polygón, modrou prvá úroveň interpolácie a zelenou druhá úroveň. Červený bod predstavuje výsledok poslednej interpolácie a súčasne bod Bézierovej krivky pre $t = \frac{1}{2}$.



OBR. 11. Jednotlivé úrovne algoritmu de Casteljau pre kubickú Bézierovu krivku a parameter $t = \frac{1}{2}$.

7.4.5. *Rozdelenie Bézierovej krivky.* Pomocné body vytvorené algoritmom de Casteljau neposkytujú iba výsledný bod krivky. Hraničné body trojuholníkovej schémy zároveň tvoria riadiace body dvoch nových Bézierových kriviek. Prvá z nich opisuje pôvodnú krivku od parametra 0 po parameter t , druhá od parametra t po parameter 1. Obe časti sú pritom znovu parametrizované na interval $[0, 1]$.

Pre Bézierovu krivku stupňa n sú riadiace body ľavej časti dané vzťahom

$$(7.23) \quad \mathbf{L}_i = \mathbf{P}_0^{(i)}(t), \quad i = 0, 1, \dots, n,$$

a riadiace body pravej časti vzťahom

$$(7.24) \quad \mathbf{R}_i = \mathbf{P}_i^{(n-i)}(t), \quad i = 0, 1, \dots, n.$$

Ľavú časť čítame po ľavom okraji interpolačnej schémy od pôvodného bodu $\mathbf{C}_0$ až po vypočítaný bod krivky. Pravú časť čítame od vypočítaného bodu po pravom okraji schémy až k pôvodnému bodu $\mathbf{C}_n$. Spoločný bod oboch častí je

$$(7.25) \quad \mathbf{L}_n = \mathbf{R}_0 = \mathbf{P}_0^{(n)}(t) = \mathbf{B}(t).$$

pôvodných súradníc x a y a koeficientov a a b . a súradnica y' bude vypočítaná z pôvodných súradníc x a y a koeficientov c a d .

$$(6.7) \quad \begin{aligned} x' &= ax + by \\ y' &= cx + dy \end{aligned}$$

čo môžeme vyjadriť pomocou maticového zápisu nasledovne:

$$(6.8) \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

kde sme rovnice vyjadrili pomocou jednej matice rozmeru 2×2 a dvoch matíc rozmeru 2×1 , teda dvoch vektorov.

Pri transformáciách geometrických dát je potrebné maximalizovať rýchlosť výpočtu a zabezpečiť tak plynulé zobrazenie meniacich sa objektov na zobrazovacom zariadení v reálnom čase. Matice sú vhodné hlavne preto, že umožňujú čiastkové transformácie vzájomne skombinovať a následne výslednou transformačnou maticou násobiť všetky vektory obsahujúce pozície bodov z ktorých získame body transformované. Je to omnoho efektívnejšie, ako na každý bod aplikovať čiastkové transformácie jednotlivo pre každú čiastkovú transformáciu. Pri potrebe niekoľko krát za sekundu transformovať v niektorých prípadoch státisíce bodov nám táto výpočtová metóda výrazne znižuje výpočtovú náročnosť.

Pri riešení čiastkových transformácií zistíme, že ak chceme pomocou jednej transformačnej matice vyjadriť kombináciu všetkých transformácií, nevystačíme si s maticou rozmeru 2×2 , ale budeme musieť použiť maticu 3×3 . Upravené vzťahy budú vyzeráť nasledovne:

$$(6.9) \quad \begin{aligned} x' &= ax + by + p \\ y' &= cx + dy + q \end{aligned}$$

čo po zavedení do matíc bude vyzeráť nasledovne:

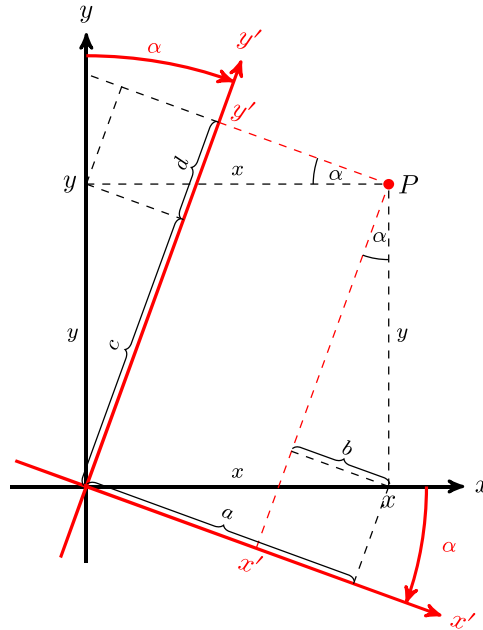
$$(6.10) \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & p \\ c & d & q \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Číslo 1, ktoré sa objavilo ako tretí člen vo vektoroch a na hlavnej diagonále štvorcovej matice je spomínaný homogenizačný faktor, pomocou ktorého sme rozšírili maticu a vektory na potrebný rozmer. Konkrétny dôvod, prečo bolo nevyhnutné rozšíriť rozmer transformačných matíc z rozmeru 2×2 na rozmer 3×3 a prečo bol s výhodou využitý homogenizačný faktor uvidíme pri prvej popísanej transformácii, ktorou je posunutie bodu v rovine.

Posunutie bodu v rovine - translácia

Translácia bodu v rovine je založená na tom, že posúvaním pôvodného bodu P do novej polohy P' meníme pôvodné hodnoty x a y o veľkosť posunutia Δx v smere osi x a veľkosť posunutia Δy v smere osi y . Táto situácia je načrtnutá na obr. č. 2.

$$(6.11) \quad \begin{aligned} x' &= x + \Delta x \\ y' &= y + \Delta y \end{aligned}$$



OBR. 44. Vyjadrenie polohy bodu v otočenom súradnicovom systéme.

Bod otáčame proti smeru hodinových ručičiek o uhol α

$$(6.18) \quad \begin{aligned} \frac{a}{x} &= \cos(\alpha) \Rightarrow a = x * \cos(\alpha); \quad \frac{b}{y} = \sin(\alpha) \Rightarrow b = y * \sin(\alpha) \\ \frac{c}{y} &= \cos(\alpha) \Rightarrow c = y * \cos(\alpha); \quad \frac{d}{x} = \sin(\alpha) \Rightarrow d = x * \sin(\alpha) \end{aligned}$$

Ďalej pre výpočet súradnice x' a y' dostaneme tieto vzťahy:

$$(6.19) \quad \begin{aligned} x' &= x * \cos(\alpha) - y * \sin(\alpha) \\ y' &= x * \sin(\alpha) + y * \cos(\alpha) \end{aligned}$$

Ďalšou možnosťou získania transformačných vzťahov pre otočenie bodu je vyjadriť polohu bodu pomocou polárnych súradníc. Táto situácia je popísaná na obrázku 5:

Pri odvodení vzťahov využijeme pravouhlý trojuholník a bod P určený polomerom r a uhlom ϕ .

$$(6.20) \quad \begin{aligned} x &= r * \cos(\phi) \\ y &= r * \sin(\phi) \end{aligned}$$

Súradnice otočeného bodu P' na základe polomeru r získame pripočítaním uhla α .