

DYDUCH Tadeusz

ZDZIARSKI Andrzej

Samodzielna Pracownia Geometrii Wykreślnej i Grafiki Inżynierskiej,  
Politechnika Krakowska

## KOMPUTEROWE WSPOMAGANIE PROJEKTOWANIA DEFORMACYJNYCH SIATEK ANAMORFICZNYCH<sup>1</sup>

**Streszczenie.** Opracowanie dotyczy zastosowania metod komputerowych w analizie właściwości geometrycznych oraz realizacji pewnego przekształcenia na płaszczyźnie. Przekształcenie to odpowiada odbiciu w zwierciadle walcowym obrotowym figur płaskich leżących w płaszczyźnie prostopadłej do tworzących walca.

## COMPUTER-AIDED DESIGN OF DEFORMATION NETS FOR THE ANAMORPHIC TRANSFORMATION

**Summary.** The paper concerns computer science methods applied to geometry features analysis and application of the special transformation. This transformation is equivalent to the reflexion in a circular cylinder surface of plane elements, while their plane is perpendicular to the axis of the cylinder mirror.

### 1. Wstęp

Często eksploatowany w malarstwie efekt lustra wody wymaga od twórcy znajomości zasad odbić zwierciadlanych. Teoria odbić zwierciadlanych jest od dawna obecna w geometrii wykreślnej. Ostatnio rozwój technik komputerowych umożliwił stworzenie programów modelujących złożone 3 - wymiarowe obiekty (np. wnętrza

---

<sup>1</sup> Opracowanie sponsorowane częściowo z grantu Komitetu Badań Naukowych nr PB 789/T07/96/10 oraz częściowo z grantu Komitetu Badań Naukowych nr PB 855/T11/95/09

mieszkalne), w których wybrane powierzchnie wykazują własności powierzchni zwierciadlanych. Popularnym programem oferującym takie możliwości jest 3D Studio firmy Autodesk. Wraz z renderingiem, to jest stopniowaniem jasności powierzchni w zależności od kąta padania promieni oświetlających, wyznaczaniem cieni oraz nakładaniem tekstur na powierzchnie brył, daje to doskonałe efekty odbioru (percepcji) obrazów obiektów 3 - wymiarowych (fotorealizm).

Funkcje te nie są jednak dobrze udokumentowane. Nadto najczęściej algorytmy wyznaczania fotorealistycznych obrazów komputerowych są bardzo złożone obliczeniowo.

Celem niniejszego opracowania jest podanie metody algorytmicznej, umożliwiającej wyznaczanie obrazów lub przeciwobrazów figur płaskich w odbiciu zwierciadlanym w powierzchni walcowej obrotowej o tworzących prostopadłych do płaszczyzn tych figur.

## 2. Ustalenia wyjściowe

Opisany poniżej algorytm opiera się na wcześniejszych pracach A. Zdziarskiego [2], dotyczących analizy geometrycznej przekształcenia anamorficzno-refleksyjno-walcowego. W szczególności opieramy się na sformułowanej tam tezie 1.

### Teza 1

Niżej opisane przekształcenie płaskie (płaszczyzny  $\alpha$  na siebie) realizuje identyczne przyporządkowanie punktów płaszczyzny  $\alpha$  sobie, jak przyporządkowanie punktów odbijanych i pozornych, obserwowanych fizycznie w odbiciu w walcowej powierzchni zwierciadlanej o tworzących prostopadłych do płaszczyzny  $\alpha$ . W tym przekształceniu płaskim kierująca powierzchnie lustrzanej na płaszczyźnie  $\alpha$  pełni rolę „krzywej zwierciadlanej”, zaś rzut prostokątny stanowiska  $Ob'$  stanowi środek obserwacji. Przekształcenie przyporządkowuje sobie punkty płaszczyzny  $\alpha$  zgodnie z regułami odbicia w zwierciadle: kąty padania i odbicia są równej miary oraz promienie padający i odbity są równej długości.

Na mocy tezy 1 można wyznaczyć pozorny (wirtualny) obraz odbity w zwierciadle walcowym w dwu etapach:

- I etap polega na wyznaczeniu obrazu anamorfy (przeciwobrazu) w wyniku przekształcenia płaskiego;
- II etap polega na wyznaczeniu rzutu środkowego ze środka obserwacji Ob, uzyskanego w I etapie obrazu.

Przedmiotem niniejszej pracy jest opracowanie metody realizacji I etapu. Trudność polega na tym, że układy równań, opisujące odwzorowanie (patrz dalej), nie są dla przypadku przekształcenia anamorfy na obraz, tj. wyznaczania obrazu pozornego, rozwiązywalne analitycznie. Zaproponowana tu metoda opiera się na wyznaczaniu rozwiązań przybliżonych, uzyskiwanych z ciągu iteracji zbieżnego do rozwiązania dokładnego.

Istnieje wiele programów wyznaczających rzut środkowy, którego reguła jest dobrze znana. Dostarczają one tzw. interface'u użytkownika - zbioru procedur umożliwiających wprowadzanie danych, zadawanie parametrów przekształceń oraz generowanie obrazów komputerowych.

Zdecydowano opracować i przetestować badany algorytm w środowisku jednego z najpopularniejszych programów CAD - AutoCAD. Umożliwiło to skupienie uwagi na samych procedurach przekształceń, bez potrzeby realizacji własnych procedur wejścia i wyjścia.

Pakiet oprogramowania AutoCAD firmy Autodesk jest wyposażony w interpreter dialektu języka LISP o nazwie AutoLISP. W dialekcie tym zdefiniowano szereg procedur i zmiennych, ułatwiających definiowanie przekształceń geometrycznych, które mogą być następnie dołączone do zbioru standardowych poleceń systemu. Popularność pakietu AutoCAD uzasadnia wykorzystanie języka AutoLISP do dokumentowania (szczegółowego opisu) algorytmów przekształceń geometrycznych.

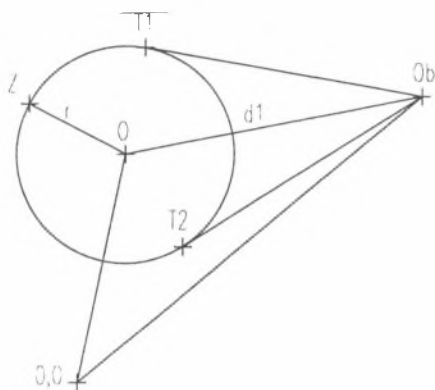
### 3. Opis algorytmu

Opis algorytmu składa się z 5 części:

- definicji podstawowych pojęć wykorzystywanych w opisie lub samym algorytmie,
- warunków przynależności punktów płaszczyzny do dziedziny odwzorowania,

- definicji zmiennych i wyrażeń wykorzystywanych do wyznaczania przekształcenia odwrotnego,
- definicji zmiennych i wyrażeń wykorzystywanych do wyznaczania przekształcenia prostego,
- opisu iteracyjnej procedury bisekcji, wyznaczającej przybliżone rozwiązania dla obydwu przekształceń.

### 3.1. Określenie elementów podstawowych, wykorzystywanych w algorytmie



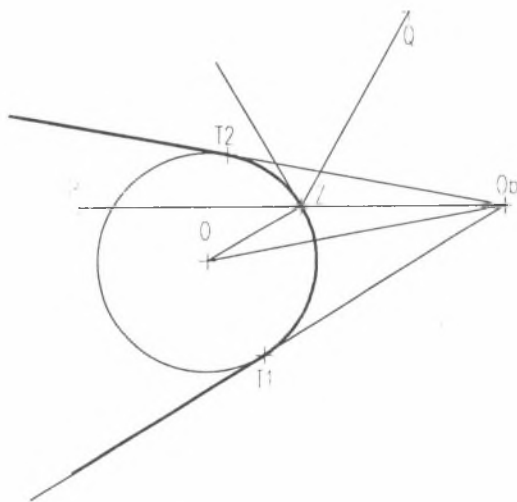
Rys. 1

Niech będzie dany na płaszczyźnie  $\alpha$  okrąg  $S(O, r)$  o środku  $O$  i promieniu  $r$ , który dalej nazywamy okręgiem odbijającym, oraz punkt  $Ob$ , zewnętrzny względem okręgu  $S$ . Odległość  $Ob-O$  oznaczmy  $d1$  i założymy  $d1 > r$ . Punkty okręgu  $S$  oznaczone  $T1$  oraz  $T2$  to punkty styczności do okręgu  $S$  prostych przechodzących przez punkt  $Ob$  (rys.1).

W cytowanej pracy A. Zdziarskiego otrzymano interesujące wyniki analizy badanego przekształcenia poprzez założenie możliwości odbić jednokrotnych w wewnętrznej (niewidocznej) powierzchni walca. Odpowiada to założeniu wykorzystania półprzezroczystej powierzchni zwierciadlanej. Przy tym założeniu także część płaszczyzny  $\alpha$  przesłonięta walcem - zwierciadłem należy do dziedziny (przeciwobrazu) przekształcenia. Założenie to, jakkolwiek wyglądające nieco sztucznie, doprowadziło do ważnego z praktycznego punktu widzenia wyniku. Pozwoliło bowiem wykazać, że

pewna rodzina okręgów, której można użyć do przyjęcia siatki ułatwiającej projektowanie anamorf, przekształca się w rodzinę łatwo wyznaczalnych ślimaków Pascala.

Dla potrzeb konstruowanego odwzorowania realizowanego komputerowo pominiemy to rozszerzenie, to znaczy realizować będziemy jedynie odbicie punktów części płaszczyzny  $\alpha$ , **nie przesłoniętej** walcem - zwierciadłem, w widocznej z punktu  $O_b$  części walca. Przy tym założeniu dziedzinę obrazu i przeciwobrazu dopełniają się na płaszczyźnie  $\alpha$ , a linia rozgraniczenia obu obszarów, składająca się z dwu półprostych stycznych do  $S$  i łuku  $S$  zawartego między nimi (patrz rys.2), jest miejscem geometrycznym punktów niezmienniczych odwzorowania, tj. przechodzących same na siebie. Konsekwentnie przyjmujemy, że miejscem geometrycznym punktów odbicia jest łuk  $S$ , przynależny do linii rozgraniczającej obszary, a odwzorowanie jest wzajemnie jednoznaczne.



Rys.2. Podział płaszczyzny na obszar przeciwobrazu - anamorfy i obrazu

### 3.2. Warunki przynależności do dziedziny odwzorowania

Na potrzeby algorytmu dokonującego przekształcenia niezbędne jest podanie ścisłej definicji przynależności punktów do dziedzin obrazu lub przeciwobrazu. Definicja ta, zapisana jako układ nierówności wektorowych, ma umożliwić algorytmiczne

sprawdzenie, czy zadany punkt można poddać żadanemu przekształceniu, tj. wyznaczyć jego obraz lub przeciwobraz w badanym odbiciu lustrzanym.

W celu zapisania tych warunków zdefiniujemy pomocniczo następujące wektory (patrz rys.2):

$\overline{ObT1}$  - łączący stanowisko obserwatora z pierwszym punktem styczności T1,

$\overline{ObT2}$  - łączący stanowisko obserwatora z drugim punktem styczności T2,

$\overline{ObO}$  - łączący stanowisko obserwatora ze środkiem okręgu zwierciadlanego,

$\overline{ObP}$  - łączący stanowisko obserwatora z badanym punktem P,

$\overline{OP}$  - łączący środek okręgu z punktem P.

Zdefiniujemy także 4 pomocnicze zmienne, określające iloczyny wektorowe powyższych wektorów.

$$b_{T1} = \overline{ObT1} \times \overline{ObO};$$

$$b_{T2} = \overline{ObT2} \times \overline{ObO};$$

$$b_{P1} = \overline{ObT1} \times \overline{ObP};$$

$$b_{P2} = \overline{ObT2} \times \overline{ObP}.$$

Obszar przeciwobrazu (anamorfy) oznaczony A możemy określić następująco, przy użyciu zdefiniowanych powyżej wektorów:

$$A \stackrel{df}{=} \{P: b_{T1}b_{P1} < 0 \vee b_{T2}b_{P2} < 0 \vee |\overline{ObP}| < |\overline{ObO}| \wedge |\overline{OP}| > r\},$$

zaś obszar obrazu B jest dopełnieniem A na płaszczyźnie:

$$B \stackrel{df}{=} \alpha \setminus A.$$

Linia rozgraniczająca oba obszary jest niezmiennikiem badanego przekształcenia (jej punkty przechodzą same na siebie).

### 3.3. Definicje zmiennych wykorzystywanych do wyznaczania przekształcenia odwrotnego

Dla wyznaczenia odwzorowania zarówno prostego jak i odwrotnego zadanego punktu kluczowe jest ustalenie punktu odbicia w „łuku odbijającym”. Znajomość punktu odbicia pozwala wyznaczyć obraz (lub przeciwobraz) poprzez odbicie w zwierciadle płaskim (na płaszczyźnie w prostej) stycznym do „krzywej zwierciadlanej” w punkcie odbicia.

Do rozwiązania zarówno przekształcenia prostego anamorfa  $\rightarrow$  obraz (pozorny), jak i odwrotnego, obraz  $\rightarrow$  anamorfa wykorzystywany jest ten sam iteracyjny algorytm bisekcji, opisany w dalszej części pracy. Algorytm ten poszukuje miejsca zerowego dla pewnego wyrażenia, którego wartość oznaczona jest „d” - innego w przypadku przekształcenia prostego, a innego w przypadku przekształcenia odwrotnego. Oba wyrażenia są tak zdefiniowane, że są one monotonicznymi funkcjami położenia punktu, przebiegającego „łuk odbijający” i przyjmują wartość 0 dla miejsca na łuku, spełniającego warunki punktu odbicia. Wartościami granicznymi d są wartości, odpowiadające początkowemu przedziałowi poszukiwań.

Dla odwzorowania odwrotnego wyrażenie d ma postać:

$$d \stackrel{\text{df}}{=} \overline{\text{ObP}} \times \overline{\text{ObZ}},$$

a jego wartości graniczne odpowiadają końcom łuku odbijającego

$$b_d = d|_{Z=T_1} = \overline{\text{ObP}} \times \overline{\text{ObT}_1}$$

$$b_g = d|_{Z=T_2} = \overline{\text{ObP}} \times \overline{\text{ObT}_2}.$$

### 3.4. Definicje zmiennych wykorzystywanych do wyznaczania przekształcenia prostego anamorfa $\rightarrow$ obraz

Dla wyznaczania przekształcenia prostego anamorfa  $\rightarrow$  obraz definiujemy wyrażenie „d” następująco:

$$d \stackrel{\text{df}}{=} \frac{\overline{\text{OZ}} \times \overline{\text{ZO}b}}{|\text{ZO}b|} + \frac{\overline{\text{OZ}} \times \overline{\text{ZQ}}}{|\text{ZQ}|},$$

gdzie:

Z - punkt łuku odbijającego,

Q - punkt, którego obrazu P szukamy.

Równanie  $d=0$  wyraża warunek równości kątów padania i odbicia (patrz rys.2). Wartościami granicznymi są w tym przypadku wartości d, obliczone dla końców łuku  $\widehat{T_1'T_2'}$  okręgu S, będącego częścią wspólną łuku odbijającego  $\widehat{T_1}$  i  $\widehat{T_2}$  i łuku okręgu S, zawartego między stycznymi, poprowadzonymi z punktu Q -  $\widehat{T_3T_4}$ .

$$\widehat{T_1'T_2'} \stackrel{\text{df}}{=} \widehat{T_1T_2} \cap \widehat{T_3T_4}$$

$$b_d = d|_{Z=T_1'} \quad b_g = d|_{Z=T_2'}.$$

Poprawne ustalenie wartości granicznych musi zapewnić istnienie dokładnie jednego miejsca zerowego wyrażenia  $d$  w funkcji położenia punktu  $Z$  na łuku odbijającym.

W przypadku przekształcenia odwrotnego z założenia wynika, iż kąt między wektorami  $\overrightarrow{ObT1}$  i  $\overrightarrow{ObT2}$  należy do przedziału  $(0, \Pi)$ . Oznacza to, że kąt między wektorem  $\overrightarrow{ObQ}$  dla punktu  $Q$  dowolnie położonego wewnątrz obszaru  $B$  a wektorem  $\overrightarrow{ObT1}$  należy do przedziału  $(-\Pi, 0)$ , zaś kąt między wektorami  $\overrightarrow{ObQ}$  i  $\overrightarrow{ObT2}$  należy do przedziału  $(0, \Pi)$ . Zatem  $d$  posiada dokładnie jedno  $0$  dla  $Z \in \widehat{T1T2}$ .

W przypadku przekształcenia prostego zachodzi:

$$d \stackrel{\text{df}}{=} \frac{\overrightarrow{OZ} \times \overrightarrow{ZOOb}}{|\overrightarrow{ZOOb}|} + \frac{\overrightarrow{OZ} \times \overrightarrow{ZQ}}{|\overrightarrow{ZQ}|} = |\overrightarrow{OZ}| (\sin \angle (\overrightarrow{OZ}, \overrightarrow{ZOOb}) + \sin \angle (\overrightarrow{OZ}, \overrightarrow{ZQ}))$$

Dla  $Z \in \widehat{T1T2} < \angle (\overrightarrow{OZ}, \overrightarrow{ZOOb}) \in \left\langle -\frac{\Pi}{2}, \frac{\Pi}{2} \right\rangle$ , natomiast dla  $Z \in \widehat{T3T4} < \angle (\overrightarrow{OZ}, \overrightarrow{ZQ})$

$$\in \left\langle -\frac{\Pi}{2}, \frac{\Pi}{2} \right\rangle.$$

Zatem dla  $Z \in \widehat{T1T2} \cap \widehat{T3T4}$  obydwa kąty zawarte są w przedziale  $\left\langle -\frac{\Pi}{2}, \frac{\Pi}{2} \right\rangle$ , w

którym funkcja  $\sin(x)$  jest monotoniczna. Równocześnie, co łatwo sprawdzić, przy ustalonym kierunku przebiegania łuku  $\widehat{T1'T2'} = \widehat{T1T2} \cap \widehat{T3T4}$  przez punkt  $Z$  oba kąty zgodnie rosną lub maleją.

Zatem  $d$  jako suma dwu zgodnie zmieniających się funkcji monotonicznych w ustalonym przedziale jest w tym przedziale monotoniczna, a więc ma najwyżej jedno miejsce zerowe. Z konstrukcji wynika, że w tym przedziale istnieje rozwiązanie. Łącznie daje to gwarancję zbieżności ciągu iteracji do rozwiązania.

### 3.5. Iteracyjna procedura bisekcji

Zastosowana procedura wyznaczania miejsca zerowego funkcji  $d(\varphi)$ , gdzie  $\varphi$  jest kątem wektora  $\overrightarrow{OZ}$  w głównym układzie współrzędnych, jest iteracyjną procedurą



bisekcji. Daje ona, przy właściwym doborze przedziału początkowego, gwarancję wyznaczenia rozwiązania o zadanej dokładności. Przyjmując liczbę iteracji  $n=30$  wyznacza ona przedział zawierający 0 o długości  $2^{-30} \approx 10^{-9}$  długości przedziału początkowego.

Procedura startuje od wyznaczenia przedziału początkowego dla kąta  $\varphi: < \varphi_d, \varphi_g >$ , gdzie  $\varphi_d$  - kąt wektora  $\overline{OT1'}$ ,  $\varphi_g$  - kąt wektora  $\overline{OT2'}$ . Następnie wyznacza się kąt wektora  $\overline{OZ}$ , gdzie punkt Z ma połović mniejszy z dwu łuków, na które dzielią okrąg S punkty T1', T2'. Dlatego po obliczeniu średniej arytmetycznej

$$\varphi' = \frac{\varphi_d + \varphi_g}{2} \text{ badany jest warunek odległości } \varphi' \text{ od } \varphi_d.$$

$$\text{Jeśli } \frac{3}{2} \Pi > |\varphi' - \varphi_d| > \frac{\Pi}{2},$$

należy przyjąć:

$$\varphi = \varphi' + \Pi, \text{ gdy } \varphi' \leq \Pi,$$

$$\varphi = \varphi' - \Pi, \text{ gdy } \varphi' > \Pi.$$

W pozostałych przypadkach przyjmuje się  $\varphi = \varphi'$ .

Dla tak obliczonego kąta wektora  $\overline{OZ}$  oblicza się wartość funkcji  $d(\varphi)$ . W następnej iteracji kątem połowiącym  $\varphi$  zastępuje się ten z kątów  $\varphi_d$  lub  $\varphi_g$ , dla którego funkcja  $d(\varphi)$  ma taki sam znak.

Procedura bisekcji zapisana w tzw. pseudokodzie wygląda następująco:

$n := 1;$

$\varphi_d := < \overline{OT1}; \varphi_g := < \overline{OT2};$

$b_d := \overline{ObP} \times \overline{ObT1}; \quad b_g := \overline{ObP} \times \overline{ObT2};$

while  $|d| > 10^{-8} \wedge n < 30$  do {

$$\varphi := \frac{\varphi_d + \varphi_g}{2};$$

$$\text{if } \left( \frac{3}{2} \Pi > |\varphi - \varphi_d| > \frac{\Pi}{2} \right) \text{ do } \{$$

if  $(\varphi \leq \Pi)$   $\varphi := \varphi + \Pi$ ; if  $(\varphi > \Pi)$   $\varphi := \varphi - \Pi$ ;}; koniec if

$Zx := Ox + r \cos \varphi; \quad Zy := Oy + r \sin \varphi;$

$d := \overline{ObP} \times \overline{ObZ};$

if  $(\text{sign } d = \text{sign } b_g)$  do {

```

    bg: = d; φg: = φ } else do { bd: = d; φd: = φ } ;
n: = n + 1;
} ; koniec while.

```

Dokumentacja algorytmu w postaci listingu w języku AutoLISP jest załączona na końcu opracowania.

## 4. Opis implementacji algorytmu zrealizowanej w środowisku AutoCAD

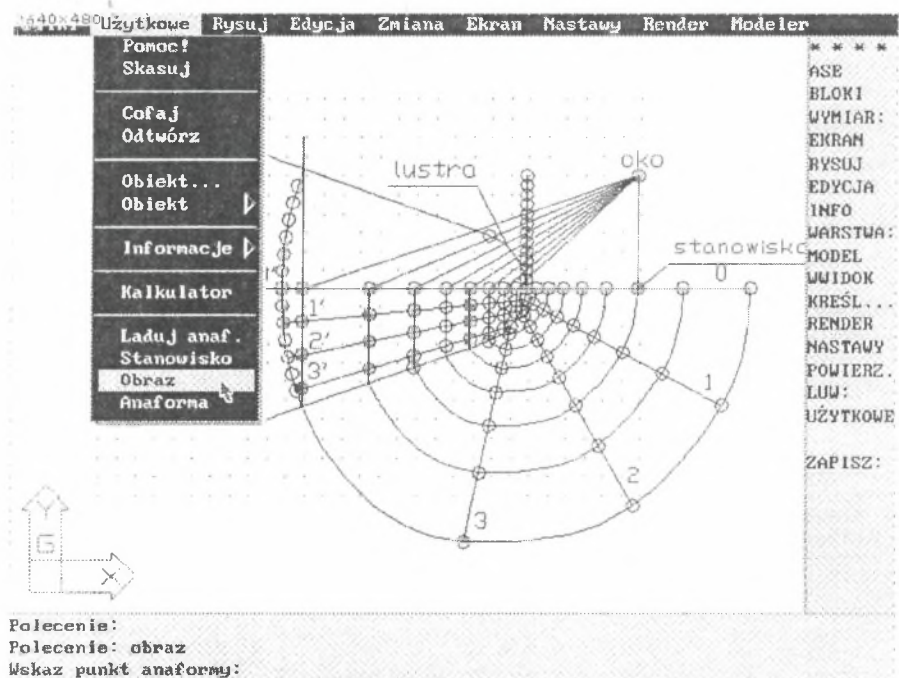
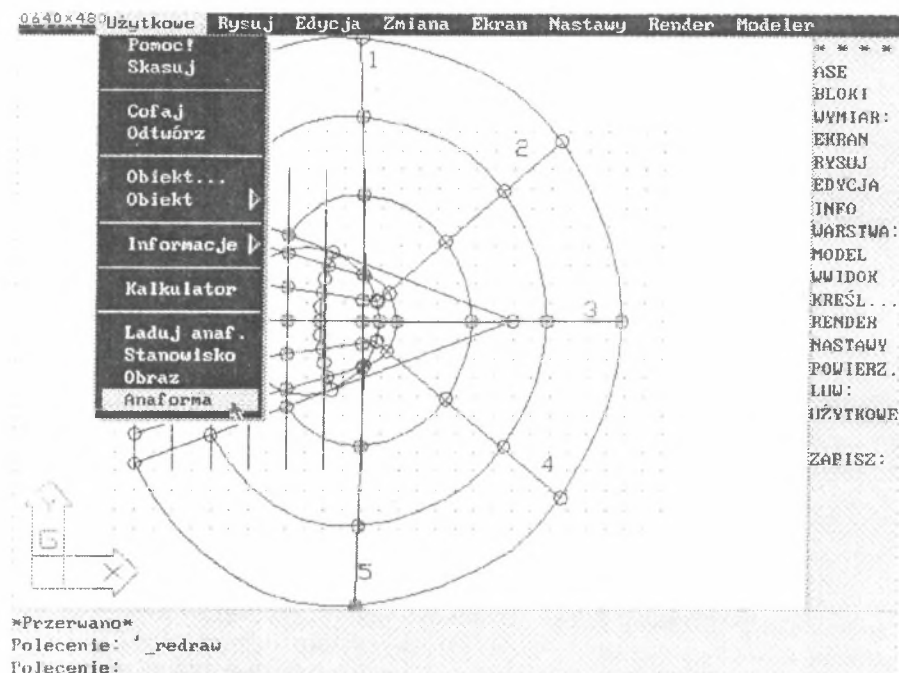
Jak już wcześniej stwierdzono, omawiany algorytm został zaimplementowany i przetestowany w środowisku AutoCAD w postaci procedur, których nazwy dołączono do zbioru standardowych komend. Są one dostępne zarówno z linii komend, jak i menu rozwijalnego (tzw. pop-up menu). Są to procedury:

- zadawania elementów określających odwzorowanie: okręgu odbijającego S oraz stanowiska obserwatora Ob;
- wyznaczania odwzorowania prostego: anamorfa → obraz dla zadanego punktu;
- wyznaczania odwzorowania odwrotnego: obraz → anamorfa dla zadanego punktu.

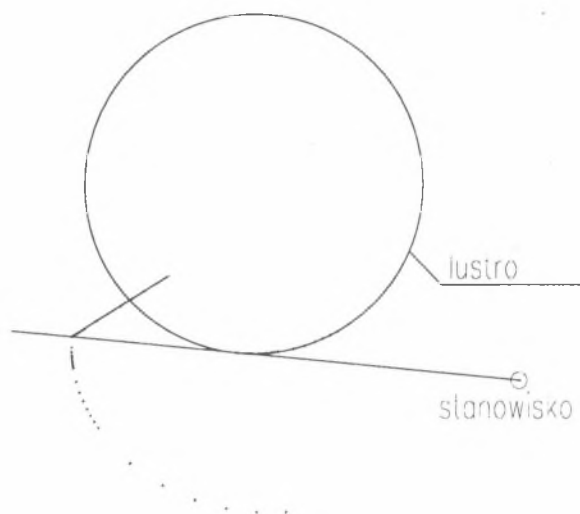
Do wykonywania pozostałych funkcji niezbędnych do stosowania przekształcenia mogą być użyte standardowe komendy systemu. W szczególności implementację wykorzystano do przebadania własności odwzorowania w pobliżu brzegu obszarów, a także do wykonania przykładowych siatek anamorficznych.

## 5. Podsumowanie

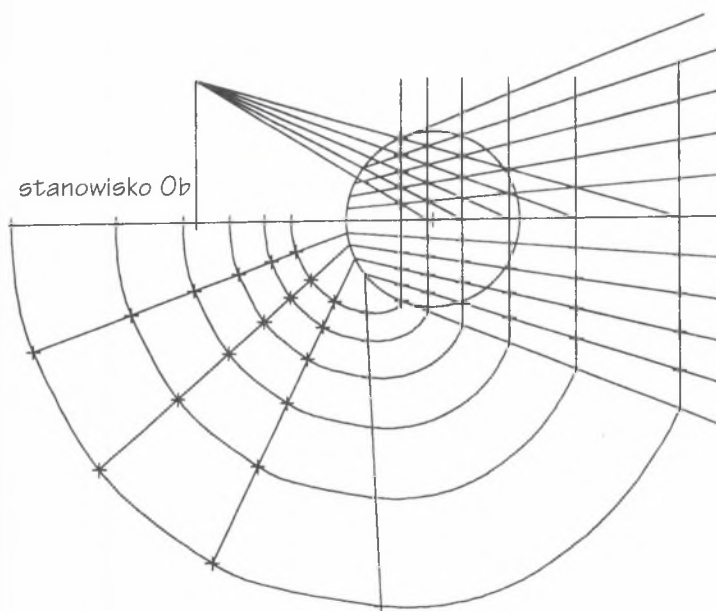
Opracowanie dotyczy zastosowania metod komputerowych w analizie własności geometrycznych oraz stosowania przekształcenia, odpowiadającego odbiciu w zwierciadle walcowym obrotowym figur płaskich, leżących w płaszczyźnie prostopadłej do tworzących walca. Zastosowane podejście łączy metody geometrii opisowej i analitycznej z typowymi dla zastosowań komputerów algorytmami obliczeń przybliżonych.



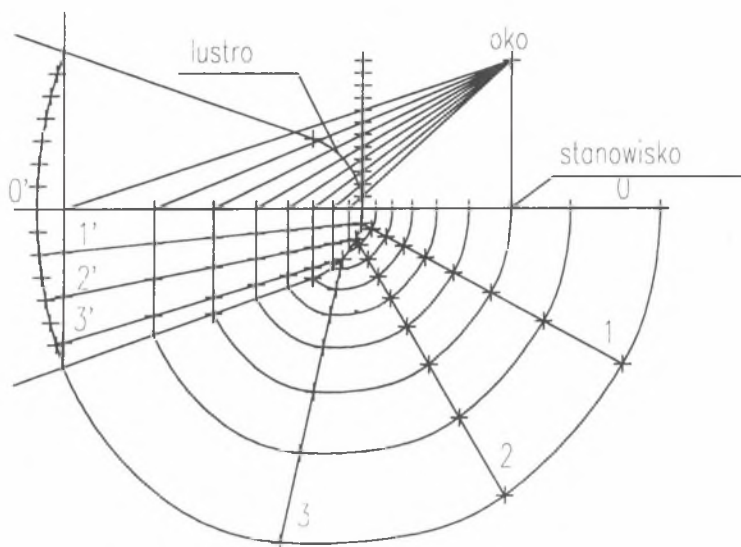
Rys.3 i 4. Obrazy ekranu AutoCAD 12 PL z zainstalowaną implementacją omawianego algorytmu.  
Widoczne dodatkowe funkcje pop-up menu



Rys.5. Przykład wykorzystania programu do analizy własności odwzorowania w pobliżu brzegu obszarów



Rys.6. Przykład konstrukcji siatki do wykreślania anamorf z wykorzystaniem opracowanej nakładki



Rys. 7. Inny przykład siatki do wykreślania anamorf

Opracowane algorytmy umożliwiają łatwe wyznaczanie w środowisku AutoCAD zarówno przeciwobrazów - anamorf, jak i obrazów odbitych zadanych anamorf dla dowolnie wybranych stanowisk i zwierciadeł. Uzyskane wyniki potwierdzają w pełni przydatność podejścia do badań problemów wizualizacji komputerowej (nazywanej czasem modelowaniem wirtualnym), łączącego klasyczne metody geometrii syntetycznej i analitycznej z metodami numerycznymi i możliwościami modelowania obecnych komputerów i programów komercyjnych.

Nowe problemy badawcze, które pojawiły się podczas opracowywania obecnego tematu, to:

- otwarte zagadnienie przekształcania zbiorów ciągłych z wykorzystaniem metod interpolacji funkcjami sklejanymi, oferowanymi przez większość programów komercyjnych CAD;
- „wirtualna” realizacja dla celów badawczych pełnego, niejednoznacznego odwzorowania, przy założeniu pojedynczego odbicia wewnętrznego i półprzepuszczalności zwierciadła.

## LITERATURA

1. DYDUCH T., ZDZIARSKI A.: Pewne algorytmy odbić lustrzanych - mat. IV Ogólnopolskiego Seminarium Geometria i Komputer - Wisła'95, Politechnika Śląska, Gliwice 1995
2. ZDZIARSKI A.: Anamorphical transformation of Space Elements Aided with the Deformation Net, Proceedings of the 7th ICECGDG, Politechnika Krakowska, Kraków 1996
3. HORCZYCZAK M., SKIERSKI J.: AutoLISP dla użytkowników AutoCAD'a, Wyd.PLJ, Warszawa 1991

Recenzent: Dr hab.inż. Jerzy Mroczkowski  
Prof. Politechniki Wrocławskiej

Dodatek: program w języku AutoLISP, realizujący funkcje przedstawione w opracowaniu.

```
(defun c:stanow() ;ustalenie polozenia lustra i obserwatora
(setvar "osmode" 0)
(setq test T)
(while test
  (setq entx (entsel "\nWskaz lustro walcowe:"))
  (setq ent (car entx))
  (setq hn (cadr entx))
  (if (equal ent nil)
    (princ "\nNic tu nie ma. Probuaj dalej.")
    (progn
      (setq entl (entget ent))
      (if (equal (cdr (assoc '0 entl)) "CIRCLE")
        (setq test nil)
        (princ "\nWskazany obiekt nie jest okregiem!"))
      )))
(setq hn (osnap hn "bliski")) ;uchwyt okregu
(setq sr (osnap hn "centrum")) ;srodek okregu
(command "punkt" sr)
(setq ob (getpoint "\nWskaz stanowisko obserwatora:"))
(command "punkt" ob)
(setq t1 (osnap hn "styczny")) ;pkt stycznosci
(command "punkt" t1)
(command "lustro" "s" "" ob sr "")
(setq t22 (entget (entlast)))
(setq t2 (cdr (assoc '10 t22))) ;drugi pkt stycznosci
(setq l1 (angle sr t1) l2 (angle sr t2))
(if (> l1 l2) (setq l l2 l2 l1 l1 l xx t1 t1 l2 l2 xx))
(setq xob (car ob) yob (cadr ob)) ;wspolrzedne obserwatora
(setq xsr (car sr) ysr (cadr sr)) ;wspolrzedne srodka walca
```

```

(setq rs (distance sr hn)) ;promien walca
(setq rob (distance sr ob)) ;odleglosc obserw. od srodka walca
(setq xt1o (- (car t1) xob)) ;skladowe wektorow obserwator-pkty stycznosci
(setq yt1o (- (cadr t1) yob))
(setq xt2o (- (car t2) xob))
(setq yt2o (- (cadr t2) yob))
(setq pi2 (/ pi 2.0))
(setq p2 (* pi 2.0))
(princ)
) ;koniec stanow

(defun sprawdz() ;sprawdzenie poprawnosci polozenia xx (za lustrem)
(setq bt1 (- (* xt1o (- ysr yob)) (* yt1o (- xsr xob))))
(setq bt2 (- (* xt2o (- ysr yob)) (* yt2o (- xsr xob))))
(setq bx1 (- (* xt1o (- (cadr xx) yob)) (* yt1o (- (car xx) xob))))
(setq bx2 (- (* xt2o (- (cadr xx) yob)) (* yt2o (- (car xx) xob))))
(setq dis1 (distance xx ob) dis2 (distance xx sr))
(if (or (> 0 (* bt1 bx1)) (> 0 (* bt2 bx2)) (and (> rob dis1) (< rs dis2)))
    (setq obraz nil) (setq obraz T)
) ;koniec sprawdz

(defun c:anaforma() ;wyznaczanie przeciwbrazu
(setq xx (getpoint "\nWskaz punkt obrazu:")) ;punkt pozorny do przeksztalenia
(sprawdz)
(if obraz (progn
    (setq xxob (- (car xx) xob)) ;skladowe wektora obs.-pkt pozorny
    (setq yxob (- (cadr xx) yob))
    (setq b1 (- (* xxob yt1o) (* yxob xt1o))) ;wartosci graniczne sprawdzania
    (setq b2 (- (* xxob yt2o) (* yxob xt2o)))
    (setq lp1 (angle sr t1) lp2 (angle sr t2) n 1 d 30)
    (while (and (not (equal d 0 0.00001)) (< n 30))
        (progn
            (setq l (/ (+ lp2 lp1) 2.0))
            (if (> (- lp2 lp1) pi) (setq l (+ l pi)))
            (if (> l p2) (setq l (- l p2)))
            (setq xpol (+ xsr (* rs (cos l))))
            (setq ypol (+ ysr (* rs (sin l))))
            (setq d (- (* xxob (- ypol yob)) (* yxob (- xpol xob))))
            (if (< 0 (* d b1)) (setq lp1 l b1 d) (setq lp2 l b2 d))
            (setq n (+ 1 n))
        )
    ) ;koniec while
    (setq zz (list xpol ypol 0))
    (command "punkt" xx)
    (command "lusto" "s" "" zz (polar zz (+ (angle sr zz) pi2) 30) "")
    )(prompt "\nPunkt musi byc wskazany poza lustrem!")
)(princ)
) ;koniec anaforma

(defun odb() ;pomocnicza dla obraz
(setq ilw1 (- (* xpols (- yob ypol)) (* ypols (- xob xpol))))
(setq ilw2 (- (* xpols (- yyy ypol)) (* ypols (- xyy xpol))))
(setq mod1 (sqrt(+ (* (- xob xpol) (- xob xpol)) (* (- yob ypol) (- yob ypol))))
(setq mod2 (sqrt(+ (* (- xyy xpol) (- xyy xpol)) (* (- yyy ypol) (- yyy ypol))))
(setq d (+ (/ ilw1 mod1) (/ ilw2 mod2)))
) ;koniec odb

(defun c:obraz() ;wyznaczanie obrazu z anaformy
(setq yy (getpoint "\nWskaz punkt anaformy:")) ;punkt do przeksztalenia
(setq xx yy) (sprawdz)
(if (not obraz) (progn

```

```

(command "punkt" xx)
(setq tp1 (osnap hn "styczny")) ;pkt stycznosci z pktu anaformy
(command "punkt" tp1)
(command "lustro" "s" "" yy sr "")
(setq t22 (entget (entlast)))
(setq tp2 (cdr (assoc 't0 t22))) ;drugi pkt stycznosci
(setq lp1 (angle sr tp1) lp2 (angle sr tp2))
(if (> lp1 lp2) (setq l lp2 lp2 lp1 lp1 l xx tp1 tp1 tp2 tp2 xx))
(if (and (< pi (- l2 l1)) (< pi (- lp2 lp1))) (progn
  (if (< l1 lp1) (setq tp1 t1))
  (if (> l2 lp2) (setq tp2 t2))
))
(if (and (> pi (- l2 l1)) (> pi (- lp2 lp1))) (progn
  (if (> l1 lp1) (setq tp1 t1))
  (if (< l2 lp2) (setq tp2 t2))
))
(if (and (> pi (- l2 l1)) (< pi (- lp2 lp1))) (progn
  (if (< l1 lp1) (setq tp2 tp1 tp1 t1) (setq tp1 tp2 tp2 t2))
))
(if (and (< pi (- l2 l1)) (> pi (- lp2 lp1))) (progn
  (if (> l1 lp1) (setq tp2 t1) (setq tp1 t2))
))
(setq xyy (car yy) yyy (cadr yy))
(setq xpol (car tp1) ypol (cadr tp1))
(setq xpols (- xsr xpol) ypols (- ysr ypol))
(oddb) (setq b1 d)
(setq xpol (car tp2) ypol (cadr tp2))
(setq xpols (- xsr xpol) ypols (- ysr ypol))
(oddb) (setq b2 d)
(setq lp1 (angle sr tp1) lp2 (angle sr tp2) n 1 d 30)
(while (and (not (equal d 0 0.00001)) (< n 30)) (progn
  (setq l (/ (+ lp2 lp1) 2.0))
  (if (< pi (- lp2 lp1)) (setq l (+ l pi)))
  (if (> l p2) (setq l (- l p2)))
  (setq xpols (- (* rs (cos l)))) ;wsp. wektora l-środek kola
  (setq ypols (- (* rs (sin l))))
  (setq xpol (- xsr xpols)) ;wsp. punktu polowiaczego
  (setq ypol (- ysr ypols))
  (oddb)
  (if (< 0 (* d b2)) (setq lp2 l b2 d) (setq lp1 l b1 d))
  (setq n (+ 1 n))
)) ;koniec while
(setq zz (list xpol ypol 0))
(command "punkt" yy)
(command "lustro" "s" "" zz (polar zz (+ (angle sr zz) pi/2) 30) "")
)(prompt "\nPunkt nie moze byc wskazany poza lustrem!")
)(princ)
) ;koniec obraz

```

## Abstract

The paper concerns computer science methods applied to geometry features analyse and application of the special transformation. This transformation is equivalent to the reflection in a circular cylinder surface of plane elements, while their plane is perpendicular to the axis of the cylinder mirror.



The research methodology applied combines methods of descriptive- as well as analytic geometry and typical numeric algorithms. The algorithms developed were implemented in AutoLISP. They enable users to easily calculate and show both reverse transformed elements - so called anamorphs and reflected images of the given anamorphs, for any chosen centre of observation and mirror.

The results obtained prove the high effectiveness of the research methodology applied on the field of computer visualisation (virtual modelling). This methodology combines classical methods of descriptive and analytic geometry with numerical methods and modelling power of today computer facilities and software.