

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW  
MERA-PIAP

Al. Jerozolimskie 202

02-222 Warszawa

Telefon 23-70-81

Ośrodek Automatykacji

074 Procesów Produkcji A

Zespół Sprzętu Cyfrowego

Główny wykonawca

Wykonawcy

dr inż. Bohdan Kontrymowicz  
mgr inż. Stanisław Marszałek  
mgr inż. Dariusz Okrasa  
mgr inż. Andrzej Zasucha

Konsultant

Nr zlecenia

RP-61

System wizyjny dla robotów IRp.

Zadanie nr 1.2

Opracowanie założeń i wybór  
algorytmów.

Zleceniodawca

CPBR 7.1 "Roboty przemysłowe"

Pracę rozpoczęto dnia

1987.05.01

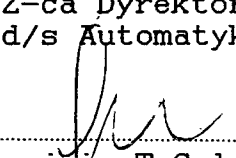
zakończono dnia 1987.09.30

Kierownik Zespołu

Z-ca Dyrektora  
d/s Automatyki

Kierownik Ośrodka

  
dr inż. B. Kontrymowicz

  
dr inż. T. Gałazka

  
mgr inż. A. Aderek

Praca zawiera:

Rozdzielnik - ilość egz:

stron 29

Egz. 1 BOINTE

rysunków 8

Egz. 2 OAP

fotografii

Egz. 3 OAE

tabel

Egz. 4 OAR

tablic 1

Egz. 5

załączników

Egz. 6

Nr rejestr. 5913

## Analiza deskryptorowa

Systemy wizyjne  
Analiza sceny  
Roboty przemysłowe  
Rozpoznawanie obrazów  
Struktura systemów wizyjnych

## Analiza dokumentacyjna

W sprawozdaniu przedstawiono założenia techniczne na system wizyjny dla robotów przemysłowych IRp. Na podstawie przeprowadzonych wcześniej studiów literaturowych dokonano wyboru algorytmów wstępnej filtracji obrazu i algorytmów rozpoznawania położenia przedmiotu we współrzędnych kamery. Algorytmy te znalazły swoje odbicie w przyjętej architekturze systemu wizyjnego.

## Tytuły poprzednich sprawozdań

Zadanie 1.1 Prace studialne.  
Nr rejestr. 5836

338.45:62/69].002.1/2 Roboty przemysłowe  
621.397.6 Urządzenie telewizyjne

UKD

PIAP-252/03-6000

## Spis treści

1.Przeznaczenie - zadania stawiane układowi wizyjnemu robota przemysłowego.....	2
2.Wymagania wynikające ze współpracy z robotem.....	4
3.Założenie na komunikację systemu z układem sterowania robota.....	9
4.Wybór algorytmów filtracji i rozpoznawania obrazów..	12
5.Architektura systemu wizyjnego dla robota.....	20
6.Programowanie systemu wizyjnego.....	27
7.Założenia na instalowanie, konserwację i serwis systemu...	27

## 1 . Przeznaczenie

Zadania stawiane przed systemem wizyjnym robota przemysłowego są nieco odmienne od typowych sytuacji w procesie rozpoznawania obrazów. Podstawową różnicą jest niezmiennosc obiektów w obrębie jednej klasy. Jest to naturalną konsekwencją produkcji seryjnej, gdzie detale jednego rodzaju muszą być identyczne, gdyż w przeciwnym razie są detalami wadliwymi, które należy wyeliminować z dalszego procesu technologicznego. Przedmiotem działań robota może być oczywiście kilka rodzajów detali, z których każdy może mieć kilka stabilnych położeń na płaszczyźnie. W sumie jednak nie powinno to być więcej niż kilkanaście różnych klas obiektów.

Następną różnicą jest sama istota zadania stawianego przed systemem wizyjnym. W większości systemów rozpoznawania obrazów celem jest zaliczenie obiektu do jednej z klas, pomimo występujących drobnych różnic i co najważniejsze dowolnego położenia w polu obrazu. Tymczasem system współpracujący z robotem ma przede wszystkim za zadanie ustalenie położenia detalu, czyli w rezultacie wyznaczenie parametrów chwytu t.j. położenia punktu  $T_p$  we współrzędnych  $x, y$  kamery oraz rotacji chwytaka wokół tego punktu. Klasyfikacja jest tu w zasadzie procesem wtórnym.

Detale występujące w procesie technologicznym mogą tworzyć sceny o różnym stopniu skomplikowania. Są to w kolejności według rosnącego skomplikowania:

- pojedynczy detal na płaszczyźnie w polu widzenia kamery;
- kilka detali wzajemnie izolowanych;
- detale na płaszczyźnie, ale stykające się ze sobą krawędziami;
- detale w stanie nieuporządkowanym (ang. bin picking).

Projektowany system przeznaczony jest dla pierwszego, najprostsze-  
go z w.w przypadków. Taki wybór jest uzasadniony następującymi  
względnymi:

- w chwili obecnej brak jest w Polsce jakichkolwiek doświadczeń w zakresie systemów wizyjnych dla robota, a zatem zaczynać należy od systemów możliwie prostych;
- obróbka sceny o większym stopniu skomplikowania wymaga większej ilości obliczeń, co albo znacznie wydłuży czas rozpoznawania, albo wymagać będzie zwiększenia mocy obliczeniowej poprzez rozbudowę układu. Żadna z tych alteratyw nie jest do przyjęcia. Rozwiązanie pierwsze oznacza obróbkę programową, a ta w najlepszym razie trwać będzie całe sekundy. Rozwiązanie drugie to rozbudowa układu, który zajmie wówczas z pewnością więcej miejsca niż dają obecnie do dyspozycji konstruktorzy układu sterowania robotem.
- scena z jednym detalem może wystąpić samoistnie np. na podajniku taśmowym lub też można ją stosunkowo łatwo zrealizować na drodze mechanicznej t.j. przez układ rynien, zderzaków i.t.p.
- dopiero badania tego względnie prostego systemu wizyjnego z robotem przemysłowym w konkretnej aplikacji stworzą możliwość opracowania realnych założeń na system wizyjny bardziej rozwinięty, o ile oczywiście będzie to uzasadnione ekonomicznie.

## 2. Wymagania wynikające ze współpracy systemu wizyjnego z robotem przemysłowym

Fakt współpracy systemu wizyjnego z robotem przemysłowym nakłada na elementy tego systemu szereg wymagań. Są to przede wszystkim:

- odporność mechaniczna i małe gabaryty kamer,
- kompatybilność elektryczna, logiczna i mechaniczna pakietów systemu ze standardem INTEL DIGIT - PROWAY zgodnie z którym wykonany jest układ sterowania robota,
- czas przetwarzania informacji odpowiedni do szybkości ruchów robota i zmian zachodzących w jego otoczeniu,
- możliwość pracy w zmiennych warunkach oświetlenia.

Wpływ powyższych wymagań na koncepcję budowy i działania systemu zostanie omówiony w kolejnych podrozdziałach.

### 2.1. Kamery.

Kamery systemu wizyjnego współpracującego z robotem przemysłowym muszą być przystosowane do zamontowania ich na ramieniu robota. Wyklucza to więc wykorzystanie kamer typu "vidikon", ze względu na ich małą odporność mechaniczną. W projektowanym systemie użyte zostaną matrycowe kamery CCD zbudowane całkowicie z elementów półprzewodnikowych, co gwarantuje dużą odporność na wstrząsy i uderzenia. Jedynym elementem kamery, który stosunkowo najłatwiej może ulec uszkodzeniu przy uderzeniu o przeszkodę jest obiektyw. Jego wymiana po ewentualnym uszkodzeniu nie jest jednak czynnością skomplikowaną.

Duży stopień scalenia układów elektronicznych gwarantuje małe gabaryty i mały ciężar kamery. Na przykład kamera CCD produkcji zakładów TESLA zawierająca matrycę 256x256 elementów ma wymiary 45x66x35 mm i waży 150g bez obiektywu. Aby zastosować tę kamerę w projektowanym systemie wizyjnym dla robota niezbędna jest jednak jej adaptacja mająca na celu umożliwienie automatycznego dopasowania do warunków oświetlenia poprzez zmianę czasu naświetlania. Zakupione gotowe kamery CCD nie mają takich możliwości ponieważ pracują w standardzie TV. Czas uzyskiwania kolejnych obrazów wynosi zatem 20 ms, co z punktu widzenia projektowanego systemu jest szybkością całkowicie wystarczającą. Pomimo konieczności przeróbki planuje się użycie tej właśnie kamery, ponieważ w chwili obecnej jest ona jedyną dostępną na rynku RWPG kamerą matrycową CCD.

## 2.2.. Pakiety systemu i komunikacja między nimi.

W kasecie układu sterowania robota, w zależności od jego konfiguracji, jest od 2 do 9 wolnych stanowisk. W przypadku pracy robota z systemem wizyjnym można przyjąć, że przynajmniej 6 stanowisk w kasecie będzie wolnych. Stwarza to możliwość umieszczenia pakietów "wizyjnych" w kasecie robota, co eliminuje konieczność umieszczania specjalnie dla systemu wizyjnego dodatkowej szafy z kasetą. Oczywiście rozwiązanie takie jest możliwe tylko w przypadku upakowania całego systemu na 6 pakietach, co według wstępnych obliczeń jest możliwe:

procesor	- 1 pakiet
pamięć	- 2 - 3 pakiety
preprocesor	- 1 pakiet
obsługa kamery	<u>- 1 pakiet</u>
	5 - 6 pakietów

Umieszczenie pakietów w kasecie robota pociąga za sobą komunikację między nimi wg. standardu INTEL DIGIT - PROWAY (norma branżowa BN - 84 3105-03).

Dołączenie do magistrali kasety układu sterowania robota jeszcze jednego pakietu procesora mogłoby zakłócać pracę zarówno układu sterowania robota jak i systemu wizyjnego. Rozwiązaniem tego problemu jest fizyczne podzielenie magistrali kasety 19" na dwie części: jednej dla układu sterowania a drugiej dla systemu wizyjnego. Można tego dokonać poprzez przecięcie linii sygnałowych magistrali pozostawiając linie zasilania. Konstruktorzy układu sterowania nie mają zastrzeżeń do takiego podziału. Umożliwia on autonomiczną pracę obu układów oraz pozwala również na umieszczenie w kasecie innych układów sensorycznych w miejscu układu wizyjnego.

Wymagania czasowe wynikające ze współpracy z robotem narzucają działanie systemu w czasie rzeczywistym. Przetwarzanie jednego obrazu nie może więc trwać dłużej niż czas aktualizacji położenia ramienia robota w trybie z korekcją trajektorii. Czas ten wg. aktualnych założeń ma wynosić 64 ms lub 256 ms w zależności od tego, czy w układzie będzie koprocessor czy nie. Jako maksymalny czas przetwarzania przyjęto 64 ms. Konieczna dokładność ustalania położenia chwytaka względem przedmiotu chwytanego waha się, w zależności od rodzaju tego ostatniego, w granicach od 1mm do 1 cm. Jeżeli przedmiot porusza się przy tym względem kamery, czyli względem ramienia robota, to w najbardziej krytycznym przypadku maksymalna prędkość nie może przekraczać 15,6 mm/s. Prędkość ta może być większa dla mniejszej dokładności wyznaczania miejsca chwytu.



### 2.3. Komunikacja między kamerami a systemem.

Wymiana informacji między kamerą (lub kamerami) a systemem odbywać się będzie za pomocą kabla transmisyjnego. Kabel będzie z jednej strony dołączony do kamery, poprowadzony wzdłuż górnego i dolnego ramienia robota aż do jego podstawy, a potem razem z kablem sterującym robota do szafy, gdzie przez otwór w jej dolnej części będzie dochodził do złącza umieszczonego w płycie czołowej pakietu obsługi kamer. Z kamery przesyłany będzie sygnał wizyjny, natomiast do kamery sygnał zegara, sygnał regulujący czas naświetlania oraz napięcia zasilające. Zadaniem pakietu obsługi kamer jest binaryzacja sygnałów wizyjnych otrzymywanych z kamer - ustawiając próg komparacji odpowiednio do obliczonego każdorazowo histogramu, przesyłanie zbinaryzowanego sygnału wizyjnego do dalszej obróbki w innych pakietach oraz wysyłanie sygnałów sterujących do kamer.

Długość kabla będzie wynosić maksymalnie 20 m, a więc odpowiednio do odległości kamer od układu sterowania robota, która standardowo nie jest większa niż 15 m. Sygnał video transmitowany będzie kablem koncentrycznym, natomiast sygnały sterujące i napięcia zasilające parami skręcanymi.

### 2.4. Oświetlenie.

Na stanowisku robota mogą występować skrajnie różne warunki "naturalnego", czyli niezależnego od systemu wizyjnego oświetlenia sceny obserwowanej przez kamerę. Oświetlenie to może zmieniać się od bardzo jasnego światła słonecznego, poprzez plamy i cienie o różnej jasności, słabe, pulsujące z częstotliwością sieci oświetlenie żarowe, aż do półmroku.

Z powodu ograniczonej dynamiki kamery i układów wstępnego przetwarzania obrazów, nadmierne różnice w oświetleniu poszczególnych części sceny mogą bardzo utrudnić, a niekiedy wręcz uniemożliwić prawidłową analizę obrazu. Nie da się tego uniknąć nawet pomimo stosowania środków zapobiegawczych np. w postaci odpowiednio dobranych algorytmów rozpoznawania oraz układu automatycznej regulacji czasu naświetlania. Pracę tego ostatniego może też poważnie zakłócać migotanie oświetlenia jarzeniowego lub żarowego.

W celu zminimalizowania szkodliwych skutków takiego oświetlenia sceny proponuję się, aby przedmioty obserwowane przez kamerę były dodatkowo oświetlone. Oświetlacz taki musi spełniać kilka warunków, a mianowicie: musi emitować światło rozproszone, najlepiej kilkupunktowe (dla uniknięcia cieni), a światło powinno mieć stałą jasność i nie może migotać.

Z tych względów najlepsze wydaje się zastosowanie zestawu składającego się z kilku reflektorów zasilanych stabilizowanym napięciem stałym np. reflektorów samochodowych. Byłyby one zamocowane na odpowiedniej głowicy wraz z kamerą

Oczywiście taka głowica oświetleniowa byłaby stosowana tylko w wyjątkowo niesprzyjających warunkach. W łatwiejszych przypadkach zakłada się użycie odpowiednich reflektorów zamocowanych na stałe np. nad robotem lub zastosowanie ekranów eliminujących wpływ zakłóceń zewnętrznych np. plam światła słonecznego padającego przez okna. Możliwe jest także użycie obu tych środków. Zakłada się, że głowica oświetleniowa z reflektorami będzie montowana tylko na ramieniu robota IRp-60, gdyż ciężar tego zestawu ograniczyłby w poważnym stopniu możliwości pracy robota IRp-6.

### 3. Założenia na komunikację systemu wizyjnego z układem sterowania robota

Komunikacja pomiędzy dwoma pracującymi autonomicznie systemami, jakimi są układ sterowania robota i system wizyjny, powinna odbywać się na poziomie decyzyjnym. Końcowym wynikiem analizy każdego obrazu będzie decyzja o zakwalifikowaniu przedmiotu obserwowanego do jednej z występujących w procesie klas obiektów oraz informacja o jego położeniu w płaszczyźnie roboczej robota.

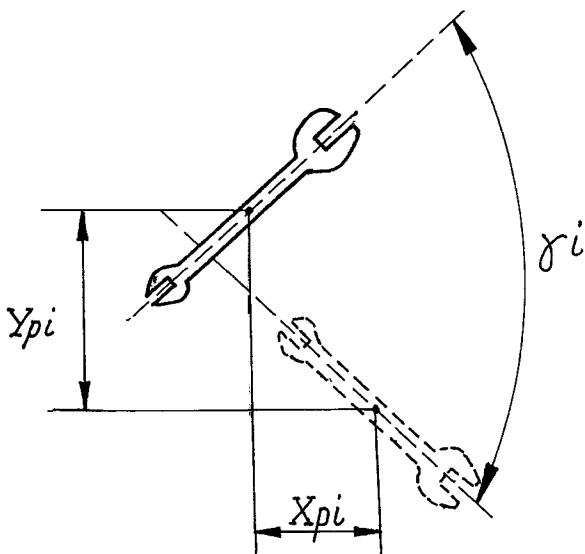
W przypadku, gdy zadaniem robota jest swobodny chwyt przedmiotu, informacja ta przybiera formę kodu transmisyjnego będącego funkcją wektora o postaci:

$$B=f'(x_{pi}, y_{pi}, \gamma_i, K_i)$$

gdzie:  $x_{pi}, y_{pi}$  - różnica współrzędnych punktu chwytu  $p$  dla  $i$ -tego przedmiotu obserwowanego i wzorca w płaszczyźnie roboczej robota,

$\gamma_i$  - kąt obrotu przedmiotu w stosunku do wzorca,

$K_i$  - składowa wektora określająca klasę do której zaliczony został  $i$ -ty przedmiot



Rysunek 1.

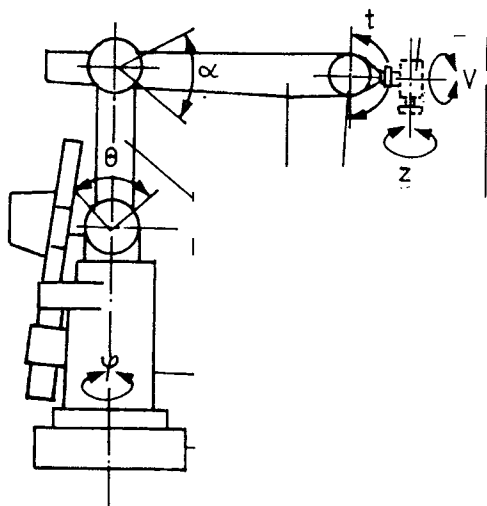
11

Wektor ten jest obliczany podczas analizy obrazu i przed przesłaniem do systemu sterowania zamieniany na kod transmisyjny.

W ramach prac rozwojowych nad oprogramowaniem robota (cel 55 CPBR 7.1) planowane jest stworzenie dla systemu sterowania nowych instrukcji, które umożliwiłyby korekcję trajektorii o przesunięcie proporcjonalne do wartości podawanego wektora korekcji w stosunku do trajektorii dowolnego kształtu oraz pozycjonowanie swobodne w zadanej płaszczyźnie roboczej.

Jednocześnie układ sterowania będzie miał możliwość przeliczania współrzędnych płaszczyzny roboczej na współrzędne wewnętrzne robota związane z jego osiami  $\varphi, \theta, \alpha, t, v$ .

- $\varphi$  - kąt obrotu osi podstaw
- $\theta$  - kąt obrotu ramienia dolnego
- $\alpha$  - kąt obrotu ramienia górnego
- t - kąt obrotu ramienia chwytaka
- v - kąt obrotu chwytaka



Rysunek 2.

Dzięki tym rozszerzeniom zestawu instrukcji możliwe będzie wykorzystanie przez układ sterowania sygnałów określających wektor korekcji obliczony we współrzędnych płaszczyzny do aktualizacji parametrów chwytu wprowadzonych do pamięci układu sterowania w trakcie uczenia się.

Najprostsze stosowane chwytaki są dostosowane do chwytania tylko jednego typu przedmiotów, tzn. że rozwarcie szczęk i siła ścisku są stałe dla wszystkich chwytów. Możliwe jest jednak także stosowanie autonomicznego układu sterowania chwytaka, który umożliwiałby zmianę siły ścisku i rozwarcia szczęk, odpowiednio do sygnałów sterujących otrzymanych z układów sensorycznych (np. wizyjnych lub dotykowych). Sygnały sterujące dla układu sterowania chwytaka byłyby obliczane na podstawie wartości składowej  $K_x$  wektora korekcji bowiem składowa ta zawiera informację o klasie, do której został zaliczony chwytany przedmiot. W przypadku stosowania chwytaka o stałych parametrach chwytu dane te byłyby niewykorzystane.

Układowo komunikacja pomiędzy systemem wizyjnym a układem sterowania robota będzie odbywała się za pomocą krótkiego kabla taśmowego zakończonego wtykami szufladkowymi. Kabel ten będzie poprowadzony wewnątrz szafy robota wzdłuż płyt czołowych pakietów umieszczonych w kasecie, a jego wtyki wetknięte będą w gniazda portów równoległych w płytach czołowych pakietów jednostek centralnych systemu wizyjnego i układu sterowania robota.

#### 4. Wybór algorytmów filtracji i rozpoznawania obrazów.

Przetwarzanie wstępne odczytanego przez kamerę obrazu realizowane jest przez układy preprocesora. Zadaniem wstępnego przetwarzania obrazu jest wyeliminowanie możliwie prostymi środkami - poza procesorem nadrzędnym - zakłóceń występujących w obrazie, powstałych w przetworniku obrazowym oraz podczas dyskretyzacji obrazu. Proces wstępnego przetwarzania musi zapewniać minimalne zniekształcenie zawartej w obrazie informacji. Przetwarzanie danych otrzymanych z kamery matrycowej (2-D) powinno być wykonywane - według założeń - w czasie rzeczywistym. Stąd też stosowane algo-rytmy filtracji muszą być szybkie i stosunkowo proste.

Wymagania czasu rzeczywistego były powodem przyjęcia koncepcji czysto hardware'owej realizacji wstępnego przetwarzania. Będzie ono wykonywane poprzez maksymalnie trzy rodzaje filtracji oraz procedurę konturowania, będącą w istocie rodzajem filtracji logicznej. Ze studiów literaturowych, głównie zaś opracowań Politechniki Wrocławskiej oraz pozycji Kam, Todd - Vision Systems wynika, że dla większości przypadków zadowalające efekty daje złożenie co najwyżej trzech niżej przedstawionych procedur filtracji. Oczywiście, w zależności od pewnych parametrów obrazu np. geometrii detalu, kontrastu detal - tło, wibracji taśmy transportowej, zadymienia itp. zagwarantowana jest możliwość doboru optymalnych dla danych warunków pracy filtracji i kolejności ich wykonywania (poprzez wymianę pamięci EPROM). Dodatkowo rozwiązania układowe zapewniają możliwość wymiany filtracji operujących na okienku 3x3 z procedurami określającymi wartość badanego punktu dla okienka 2x2 np. operatorem gradientowym krawędzi Robertsa czy też procedurami operującymi na

sąsiednich pikslach jednej linii - bez konieczności zmian sprzętowych.

Podstawowe metody filtracji realizowane są na okienku 3x3 (rys.3). Polegają one na tym, że wartość środkowego punktu okienka jest funkcją pozostałych ośmiu otaczających go punktów.

a	b	c
d	e	f
g	h	i

Rysunek 3.

Zmodyfikowana wartość punktu e określana jest zatem w oparciu o wartości punktów a,b,c,d,e,f,g,h,i.

W projektowanym systemie wizyjnym przewiduje się zastosowanie następujących procedur filtracji: dwóch logicznych i jednej arytmetycznej.

Pierwsza filtracja logiczna L1 określana jest wzorem:

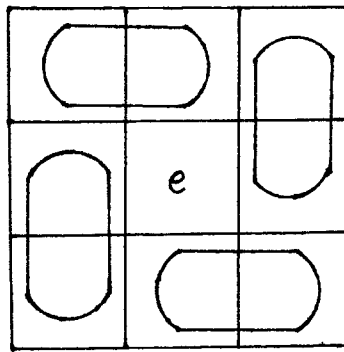
$$e' = e(a+b+c+d+f+g+h+i) + \bar{e}(abcd fghi)$$

Zgodnie z podanym wzorem punkt e' przyjmuje wartość 0, gdy wszystkie sąsiednie punktu mają również wartość 0, oraz e=1 w sytuacji, gdy sąsiednie punkty mają taką wartość. W pozostałych przypadkach wartość punktu e nie ulega zmianie. Przedstawiona wyżej filtracja eliminuje zakłócenia w postaci izolowanych punktów.

Druga filtracja logiczna określona jest wzorem:

$$e' = e(ab+cf+hi+dg) + e(a+b)(c+f)(h+i)(d+g)$$

Wykorzystanie tej metody prowadzi do zmiany wartości punktu  $e$  z 0 na 1 jeżeli w każdym z wyróżnionych sektorów (rys.4) występuje przynajmniej jedna jedynka oraz z 1 na 0, gdy w każdym sektorze znajduje się 0; w pozostałych przypadkach wartość  $e$  nie ulega zmianie.



Rysunek 4

Metoda ta pozwala wyeliminować zakłócenia w postaci izolowanych linii nie zmieniając przy tym geometrii sylwetki.

Filtracja arytmetyczna realizowana jest w oparciu o funkcję:

$$e' = \begin{cases} 1, & \text{gdy liczba jedynek w otoczeniu punktu } e \\ & \text{jest większa od } n \\ e, & \text{gdy liczba tych jedynek jest równa } n \\ 0, & \text{w pozostałych przypadkach} \end{cases}$$

Przyjmując np.  $n=4$  eliminujemy zakłócenia mniejsze od



5 punktów. Dzieje się to jednak kosztem zmiany obrysu sylwetki - występują zaokrąglenia narożników.

Konturowanie obrazu przewiduje się dokonać w oparciu o metodę czterospójności określoną wzorem:

$$e = \overline{ebdfh}$$

Oznacza to, że punkt  $e$  nie może być elementem konturu, jeżeli punkty  $b, d, f, h$  przyjmują wartość 1. Punkt  $e$  jest natomiast elementem konturu jeżeli:

- element centralny  $e=1$
- alternatywa elementów  $b, d, f, h$  jest mniejsza od 4 i większa od 1
- punkty z otoczenia punktu  $e$  o wartości 1 muszą być sąsiednie

Punkty konturu zostają następnie zapisane do pamięci w kodzie Run-Length.

#### 4.2 Rozpoznawanie obrazu. Określenie położenia i orientacji przedmiotu.

Zadaniem rozpoznawania jest zaliczenie obiektu, którego obraz dostarczył czujnik wizyjny do jednej ze znanych klas lub stwierdzenie, że obiekt jest nieznanym. Sposób realizacji rozpoznawania jest uzależniony od sposobu zakodowania obrazu i związanego z tym sposobem zdefiniowania klas obiektów. Kodowanie obrazu polega na określeniu tych jego parametrów, które są konieczne i wystarczające do rozpoznania (sklasyfikowania) obiektu oraz wyznaczenia jego położenia i orientacji. Wymagania czasu rzeczywistego stwarzają konieczność doboru odpowiednio szybkich algorytmów obliczania tych parametrów oraz realizacji sprzętowej tych algorytmów, dla których

takie rozwiązanie nie wpłynie ujemnie na elastyczność systemu wizyjnego (tzn. na możliwość zastosowania innych metod prowadzących do klasyfikacji przedmiotu, określenia jego położenia i orientacji, gdyby takie okazały się bardziej skuteczne). Z tych względów przewiduje się wykorzystanie parametrycznej metody kodowania konturów, natomiast metoda pierścieniowa zostanie użyta do określenia obrotu przedmiotu względem położenia przyjętego jako bazowe. Oczywiście rozwiązania układowe systemu nie będą wykluczać zastosowania innych metod kodowania np. metody funkcyjnej czy metody funkcji krzywizny.

#### METODA PARAMETRYCZNA.

Metoda ta polega na przypisaniu konturowi wektora wartości wybranych parametrów. Warunkiem wyboru parametrów jest ich niezależność od pozycji i orientacji obiektu w polu widzenia kamery. Przyjęto następujący podstawowy zestaw parametrów:

$$\begin{aligned}
 \text{- pole obiektu: } p_1 &= \sum_{i=1}^N \sum_{j=1}^M h_{i,j} & ; & \quad h_{i,j} = \begin{cases} 1, & \text{gdy element} \\ & \text{należy do} \\ & \text{obszaru} \\ & \text{obektu} \\ 0, & \text{w przeciwnym} \\ & \text{przypadku} \end{cases} \\
 \text{- obwód: } p_2 &= \sum_{i=1}^N \sum_{j=1}^M h_{i,j} & ; & \quad h_{i,j} = \begin{cases} 1, & \text{gdy element} \\ & \text{należy do} \\ & \text{konturu} \\ 0, & \text{w przeciwnym} \\ & \text{przypadku} \end{cases}
 \end{aligned}$$

-minimalny moment bezwładności:

$$p_3 = \frac{1}{2}(I_{xc} + I_{yc}) + \frac{1}{2} \sqrt{(I_{xc} + I_{yc})^2 + 4 \cdot I_{xcyc}}$$

$$\text{gdzie: } I_x = \sum_{i=1}^N \sum_{j=1}^M j^2 \cdot h_{i,j} \quad \text{- moment względem osi } x$$

$$I_y = \sum_{i=1}^N \sum_{j=1}^M i^2 \cdot h_{i,j} \quad - \text{moment względem osi } y$$

$$I_{xy} = \sum_{i=1}^N \sum_{j=1}^M i \cdot j \cdot h_{i,j}$$

$$I_{xc} = I_x - p_1 \cdot Y_c^2$$

$$I_{yc} = I_y - p_1 \cdot X_c^2$$

$$I_{xcyc} = I_{xy} - p_1 \cdot X_c \cdot Y_c$$

- środek ciężkości:

$$X_c = \left( \sum_{i=1}^N \sum_{j=1}^M i \cdot h_{i,j} \right) / p_1 \quad - \text{współrzędna } x \text{ środka ciężkości}$$

$$Y_c = \left( \sum_{i=1}^N \sum_{j=1}^M j \cdot h_{i,j} \right) / p_1 \quad - \text{współrzędna } y \text{ środka ciężkości}$$

- maksymalny moment bezwładności:

$$p_4 = \frac{1}{2} (I_{xc} + I_{yc}) + \frac{1}{2} \sqrt{(I_{xc} + I_{yc})^2 + 4 I_{xcyc}}$$

- minimalna odległość środka ciężkości od konturu:

$$p_5 = \min_{h_{i,j}} \sqrt{(X_c - i)^2 + (Y_c - j)^2}; \quad h_{i,j} = \begin{cases} 1, & \text{gdy element} \\ & \text{należy do} \\ & \text{konturu} \\ 0, & \text{w przeciwnym} \\ & \text{przypadku} \end{cases}$$

- maksymalna odległość środka ciężkości od konturu:

$$p_6 = \max_{h_{i,j}} \sqrt{(X_c - i)^2 + (Y_c - j)^2}$$

- średnia odległość środka ciężkości od obwodu:

$$p_7 = \sqrt{(X_c - i)^2 + (Y_c - j)^2} / p_2$$

Metodę parametryczną dzielimy na dwie fazy: treningową i operacyjną. W fazie treningowej obliczane są wzorcowe wartości parametrów dla każdego z możliwych obiektów i ich stabilnych położenia. W fazie operacyjnej - dla obiektu będącego aktualnie w polu widzenia kamery wyliczane są te parametry, a następnie porównywane z tablicą parametrów wzorcowych.

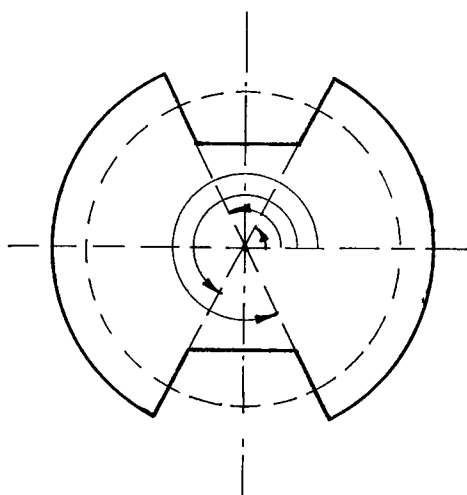
Metodą klasyfikacji jest algorytm "najbliższego sąsiada".

#### METODA PIERŚCIENIOWA.

W fazie uczenia się na obraz obiektu nakłada się okręgi o środku w środku ciężkości sylwetki, a punkty przecięcia się okręgów z konturem traktowane są jako punkty charakterystyczne.

Promienie należy dobrać tak aby:

- R był możliwie duży (większa dokładność pomiaru)
- punkty przecięcia dawały możliwie jak najwięcej informacji o położeniu obiektu
- ilość punktów przecięcia nie była zbyt duża
- punkty przecięcia nie powinny pojawiać się lub znikać przy małych zmianach promienia
- kątowne położenie punktów przecięcia nie powinno się zbyt szybko zmieniać wskutek małych zmian promienia.



Rysunek 5

Ponieważ metoda pierścieniowa realizowana jest programowo, co pociąga za sobą długi czas jej wykonywania, będzie ona wykorzystywana tylko w przypadkach klasyfikacji, gdzie metoda parametryczna nie będzie wystarczająco skuteczna.

## 5. Architektura systemu wizyjnego dla robota.

W skład systemu wizyjnego z kamerą matrycową o rozdzielczości 256x256 punktów dla robota przemysłowego wchodzić będzie 4 lub 5 pakietów o wymiarach podwójnej eurokarty. Będą się one komunikować między sobą poprzez magistralę Inteldigit-Proway. Przewiduje się wykorzystanie gotowych pakietów mikroprocesorowego systemu automatyki kompleksowej Inteldigit-Proway: pakietu MM86 z procesorem 16-bitowym 8086, który będzie pełnił w systemie wizyjnym funkcje procesora nadrzędnego oraz 1 lub 2 pakiety ML50. Pakiet ML50 zawiera pamięć programu o pojemności 16Kx16 bitów oraz pamięć danych o pojemności 4Kx16 bitów. Pamięć programu typu EPROM będzie również wykorzystywana jako pamięć wzorów okręgów dla wybranych promieni (dane dla metody pierścieniowej). Podczas inicjalizacji systemu wizyjnego dane te będą przepisywane do pamięci RAM na tym samym pakiecie, a następnie wykorzystywane przez odpowiednie algorytmy klasyfikacji obiektów.

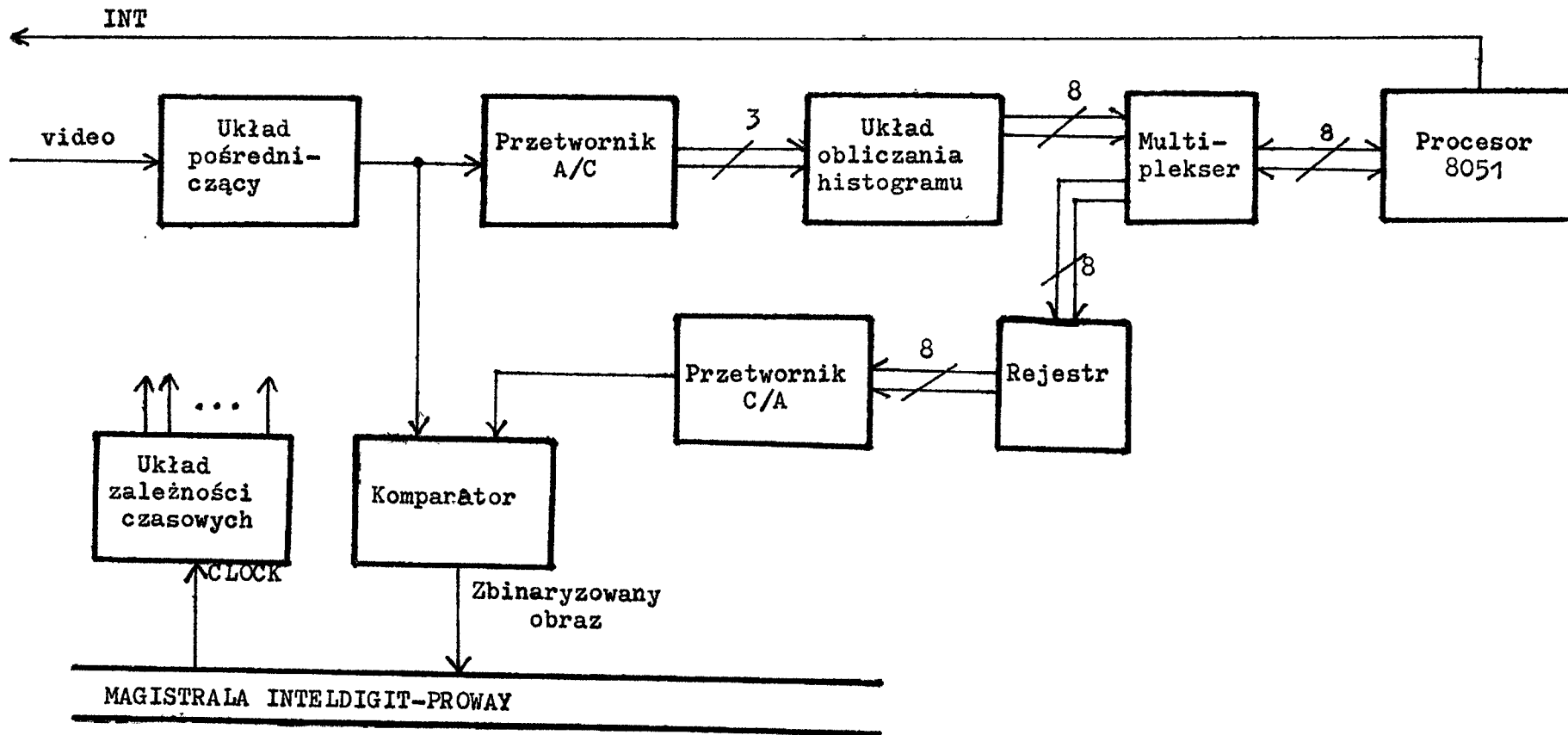
Dwa pozostałe pakiety systemu wizyjnego to:

- pakiet obsługi kamery
- pakiet preprocesora.

Pakiet obsługi kamery przedstawiony jest na rys.6.

Analogowy sygnał wizyjny (video) otrzymany z kamery CCD podawany jest na układ pośredniczący. Ma on za zadanie eliminację z sygnału video impulsów synchronizacji oraz dopasowanie sygnału użytkowego kamery do parametrów przetwornika A/C.

Ze względu na wymaganą wysoką niezawodność działania całego systemu będzie się dążyć do zastosowania 4-bitowego monolitycznego przetwornika A/C wykorzystującego metodę bezpośredniego przetwarzania równoległego ("flash").



Rys. 6 Pakiet obsługi kamery.

03

Przetworniki takie gwarantują czas przetwarzania w granicach 20 ns.

Układ wyznaczania histogramu określa rozkład liczby punktów o jednakowej jasności obrazu. Celem obliczania histogramu jest automatyczna regulacja kamery poprzez zmianę czasu naświetlania lub/i dobór progu komparacji. Histogram jest wyznaczany w trakcie trwania jednego obrazu, przy czym zliczana jest liczba punktów w 8 poziomach szarości. Analiza toru przetwarzania sygnału wykazała, że w projektowanym systemie nie jest celowe rozróżnianie większej liczby poziomów szarości niż 8, gdyż spowodowałoby to tylko nadmierną w stosunku do potrzeb rozbudowę układu. Ze wstępnych rozważań wynika, że dla ustalenia czasu naświetlania i progu komparacji 8 poziomów szarości obrazu jest wystarczające. W tej sytuacji uzasadnione wydaje się zastosowanie przetwornika A/C 3-bitowego.

Jak już wspomniano, ze względów niezawodności systemu należałoby użyć przetwornik monolityczny, ale na rynku dostępne są najczęściej przetworniki 6-bitowe. Koszt takich przetworników jest dosyć wysoki, tak że ich wykorzystanie przy jednoczesnym użyciu tylko 3 bitów byłoby ekonomicznie nie uzasadnione. Jeżeli zatem nie będzie dostępny szybki przetwornik A/C 4-bitowy, jedynie uzasadnionym rozwiązaniem wydaje się budowa przetwornika z elementów dyskretnych.

Obsługa histogramu - określenie wartości progu komparacji oraz czasu integracji obrazu w kamerze na podstawie rozkładu jasności obrazu - dokonywana jest przez mikrokontroler 8051. Program obsługi zapisany jest w jego wewnętrznej pamięci EPROM.

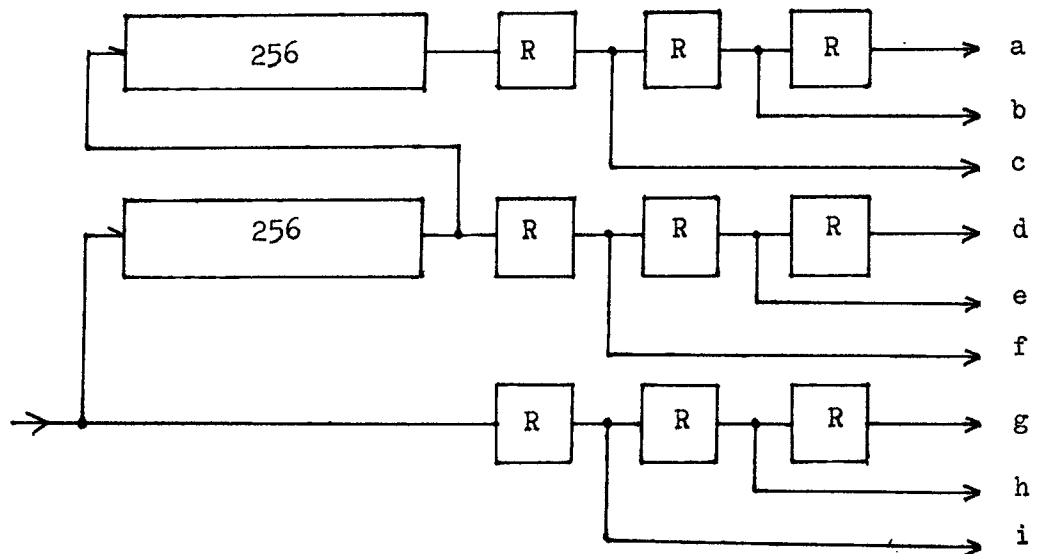
Programowo ustalony poziom napięcia odniesienia dla komparatora wpisywany jest w postaci 8-bitowego słowa do rejestru, a następnie podawany do przetwornika C/A. Napięcie wyjściowe przetwornika C/A (stanowiące próg komparacji) jest podawane na wejście komparatora, gdzie jest porównywane z sygnałem analogowym. Komparator dokonuje binaryzacji obrazu. Zbinaryzowany obraz jest przesyłany po-



przez magistralę Inteldigit-Proway do pakietu preprocesora. Obecny na pakiecie układ zależności czasowych generuje odpowiednie przebiegi czasowe w oparciu o sygnał zegarowy z pakietu MM86.

Schemat blokowy preprocesora przedstawiony jest na rys.7.

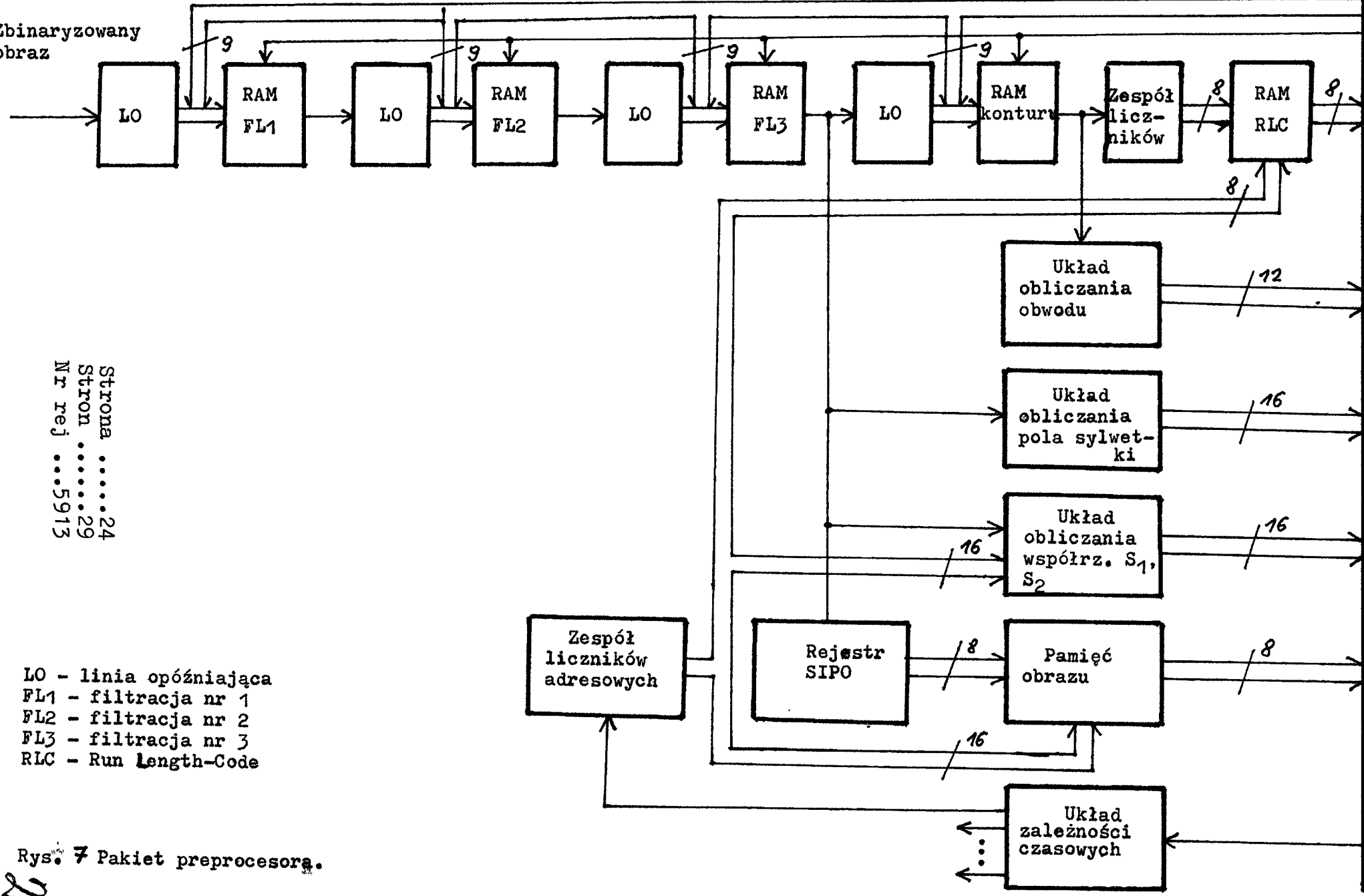
Zbinaryzowany obraz z pakietu obsługi kamery poddawany jest trzem następującym po sobie filtracjom. Maski tych filtracji przepisywane są do pamięci RAM w fazie inicjalizacji systemu wizyjnego z pamięci EPROM pakietu ML50. Ponieważ filtracje te operują na okienkach 3x3 musi zostać zapewnione przygotowanie odpowiedniej sekwencji punktów obrazu. Dokonują tego trzy linie opóźniające zgodnie z rys.8.



Rysunek 8.

MAGISTRALA INTELIGIPIPROWAY

Zbinyaryzowany obraz



Strona .....24  
Stron .....29  
Nr rej .....5913

- LO - linia opóźniająca
- FL1 - filtracja nr 1
- FL2 - filtracja nr 2
- FL3 - filtracja nr 3
- RLC - Run Length-Code

Rys. 7 Pakiet preprocesora.

26

Dziewięć punktów okienka 3 x 3 jest podawanych w formie słowa 9-bitowego na wejścia adresowe pamięci RAM o pojemności 0,5K x 1 bit (po jednej takiej pamięci dla każdej filtracji). Dla danego adresu informacją wyjściową z pamięci jest zmodyfikowana wartość środkowego punktu okienka.

Zawartością pamięci RAM konturu jest maska filtracji wydzielająca z obrazu kontur detalu. Punkty konturu sumowane są w układzie obliczającym długość obwodu. Obwód detalu jest jednym z parametrów służących do jego klasyfikacji. Adresy punktów należących do konturu, określane przez liczniki adresów, zapisywane są do pamięci w kodzie Run-Length. Punkty te stanowią dane dla algorytmów obliczających parametry chwytu dla robota (metoda pierścieniowa). Procesor nadrzędny odczytuje te dane poprzez bufory magistrali.

Układ obliczania pola sylwetki sumuje wszystkie należące do niej punkty. Wartość ta jest parametrem służącym do sklasyfikowania detalu oraz do wyznaczenia położenia środka ciężkości. Dane dla tego układu stanowi obraz po wykonanych procedurach filtracji. Te same dane wykorzystuje układ obliczający parametry S1 i S2 środka ciężkości, gdzie:

1  $X_c = S_1 / p_1$  - współrzędna x środka ciężkości

2  $Y_c = S_2 / p_1$  - współrzędna y środka ciężkości

3  $S_1 = \sum_{i=1}^N \sum_{j=1}^M i \cdot h_{i,j}$

4  $S_2 = \sum_{i=1}^N \sum_{j=1}^M j \cdot h_{i,j}$

$$; h_{i,j} = \begin{cases} 1 - \text{gdy punkt należy do obszaru obiektu} \\ 0 - \text{w przeciwnym przypadku} \end{cases}$$

Na podstawie wzorów 1 i 2 procesor 8086 będzie obliczał współrzędne środka ciężkości X i Y wykorzystując rozkaz dzielenia liczby 16-bitowej przez liczbę 8-bitową. Obciążenie zatem procesora tą częścią obliczania środka ciężkości, w której należy wykonywać dzielenie, znacznie zredukuje część sprzętową.

Dane z układów obliczających pole, obwód oraz parametry S1 i S2 środka ciężkości zapisywane będą w buforach, do których procesor nadrzędny będzie miał dostęp poprzez magistralę.

Pamięć obrazu RAM o pojemności 8Kx8 bitów przechowuje kolejne obrazy po wykonanych operacjach filtracji. Dane do zapisu natomiast przygotowywane są w postaci słów 8-bitowych przez rejestr szeregowo-równoległy typu SIPO, a wejścia adresowe pamięci obrazu sterowane są z liczników adresów. Układ zależności czasowych generuje odpowiednie przebiegi czasowe w oparciu o sygnał zegarowy z pakietu MM86.

## 6. Programowanie systemu wizyjnego dla robota.

W odniesieniu do systemu wizyjnego robota typowy podział na oprogramowanie systemowe i użytkowe jest bardzo trudny. Wynika to z faktu, iż rola użytkownika w programowaniu systemu sprowadza się jedynie do wyboru najwłaściwszej sekwencji filtracji oraz długości promieni w metodzie pierścieniowej. Z drugiej zaś strony ewentualna potrzeba nowego doboru tych parametrów przy zmianie asortymentu detali wymaga użycia specjalnych narzędzi (sprzętowych i programowych), które pozwalają na wizualizację i analizę wyników działania poszczególnych procedur i ich sekwencji. Narzędzia te podrażałyby niewątpliwie koszty systemu. Ponadto nie można od użytkownika wymagać znajomości procedur przetwarzania obrazów, co przy instalowaniu systemu jest niezbędne.

Z wymienionych wyżej względów zakłada się, że we wstępnej fazie aplikacji pilotażowych wszelkie zasadnicze zmiany asortymentu rozpoznawanych detali wymagać będą konsultacji konstruktorów systemu lub odpowiednio przeszkolonego serwisu. Ustalone w wyniku badań nowe parametry systemu będą wprowadzane przez wymianę pamięci EPROM lub jeśli pozwoli na to pojemność pamięci stałej, to za pomocą mikroprzełączników na pakiecie.

W dalszej przyszłości możliwe będzie wyposażenie systemu na stałe we wspomniane wyżej narzędzia. Będzie to jednak wymagać dość dużych zmian i uzupełnień układu sterowania robota tak pod względem sprzętowym (monitor ekranowy, pamięć obrazu, znacznie zwiększona pojemność pamięci RAM w systemie, interfejs układu wizualizacji przetwarzanego obrazu) jak i programowych (nowe instrukcje robota, możliwość wyboru parametrów systemu wizyjnego z panelu programowania robota). Na obecnym etapie rozwoju układu sterowania robota IRp uzupełnienia tego typu w terminie do planowanego terminu zakończenia pracy wydają się być zupełnie nierealne

W zakresie oprogramowania typowo systemowego, a więc poszczególnych procedur, zostaną ułożone programy dla mikrokontrolera 8051 i procesora 8086 jako procesora nadrzędnego w tym systemie.

Rola procesora 8051 sprowadzać się będzie do określania wartości progu komparacji w celu zbinaryzowania obrazu na podstawie histogramu oraz do obsługi kamery (określanie czasu integracji w zależności od zmieniającego się oświetlenia sceny).

Programy zapisane będą w wewnętrznej pamięci EPROM procesora i będą inicjalizowane przerwaniem.

Zadania procesora 8086:

- inicjalizacja systemu wizyjnego
- przepisanie masek filtracji i konturowania z EPROM'ów do odpowiednich pamięci RAM
- obliczenie środka ciężkości na podstawie przygotowanych sprzętowo danych
- w procesie uczenia się zapisanie do pamięci RAM koniecznych do późniejszego rozpoznawania parametrów przedmiotów
- rozpoznawanie przedmiotu na podstawie danych z procesu uczenia się oraz obliczonych sprzętowo parametrów
- obliczenie parametrów chwytu: wektora translacji pomiędzy położeniem przyjętym za bazowe a aktualnym środkiem ciężkości oraz kąta obrotu w stosunku do wzorca
- obsługa portu równoległego do komunikacji z robotem.

## 7. Założenia na instalowanie, konserwację i serwis systemu.

Instalowanie systemu musi się odbywać pod kontrolą projektantów systemu wizyjnego lub odpowiednio przeszkolonych pracowników serwisu. Jest to konieczne ze względu na dość dużą wrażliwość systemu na zmianę warunków otoczenia. Chodzi tu głównie o oświetlenie tj. występowanie nie zawsze możliwych do przewidzenia na etapie projektowania stanowiska cieni, odbłasków i zakłóceń pochodzących od wyładowczych źródeł światła umieszczonych w jego pobliżu. Konieczne jest też zwrócenie uwagi na możliwość wystąpienia drgań ekspozycji, jej ewentualnego zapylenia lub zabrudzenia.

Właściwe instalowanie systemu wizyjnego robota odbywać się będzie w drodze badań efektywności różnych sekwencji filtracji w rzeczywistych warunkach otoczenia. O niezbędnych do tego celu narzędziach wspomniano już w poprzednim rozdziale.

Konserwacja systemu polegać będzie głównie na utrzymaniu w czystości obiektywu kamery. Pozostałe czynności konserwacyjne dotyczące układu przetwarzania obrazu znajdującego się w kasecie szafy sterującej są identyczne jak dla układu sterowania robota.

Serwis systemu sprowadzać się będzie w razie awarii do wymiany kamery lub uszkodzonego pakietu. Naprawa odbywać się będzie w większości przypadków w warunkach laboratoryjnych, po czym możliwa będzie ponowna wymiana na podzespół oryginalny.

## System wizyjny 2-D dla robota przemysłowego.

Przeznaczenie: określenie dla robota przemysłowego parametrów chwytu pojedynczych detali znajdujących się na płaszczyźnie w polu widzenia kamery.

### Założenia techniczne:

- kompatybilność elektryczna, logiczna i mechaniczna pakietów systemu ze standardem INTEL DIGIT-PROWAY
- czas przetwarzania informacji: 64 ms
- rozdzielczość kamery matrycowej CCD: 256x256
- liczba poziomów szarości: 8
- częstotliwość taktu zegara: 8 MHz
- pamięć obrazu: 8Kx8 bitów
- liczba pakietów systemu: 4 ÷ 5
- liczba wykonywanych filtracji dla jednego obrazu: 1 ÷ 3
- metody rozpoznawania obrazów: parametryczna i pierścieniowa
- zapis zbinaryzowanego obrazu do pamięci w kodzie Run-Length.