Dekompozycja systemu sterowania robota-kasjera

Tomasz Kornuta, Cezary Zieliński
Instytut Automatyki i Informatyki Stosowanej Politechniki Warszawskiej

Streszczenie: W artykule przedstawiono formalną metodę specyfikacji systemów sterowania na przykładzie robota kasjera. Robot ten wykorzystuje aktywną wizję do realizacji swoich zadań. Aktywna wizja polega na wykonywaniu ruchów kamerą w celu aktywnego pozyskiwania informacji o otoczeniu. W tym konkretnym przypadku rozpoznawane są kody kreskowe na opakowaniach znajdujących się na ladzie.

Słowa kluczowe: robot kasjer, system sterowania, zachowanie, aktywna wizja

Zidentyfikowanie obiektu w przypadku gdy znajduje się on w dużej odległości lub jest częściowo przysłonięty przez inne przedmioty często okazuje się niemożliwe. Wprowadzenie ruchomego obserwatora (np. kamery zamontowanej na robocie) umożliwia przezwyciężenie tych trudności. Podejście to zwane jest aktywną wizją (ang. *active vision*) [1] oraz stanowi podklasę problemu aktywnego czucia (ang. *active perception*) [2], w którym informacja o otoczeniu uzyskiwana jest poprzez zmianę położenia różnego rodzaju receptorów. Ogólna definicja tego podejścia określa je jako sekwencyjny oraz interaktywny proces selekcji i analizy poszczególnych elementów sceny [5].

U zwierząt większość procesów poznawczych, w tym spostrzeganie, jest związane z aktywnością eksploracyjną o charakterze motorycznym [8]. Najprostszym przykładem takiej aktywności jest ruch gałek ocznych, podczas którego zmieniany jest region zainteresowania, co było inspiracją pracy [6]. Opisano tu sposób zmiany położenia kamery, odwzorowujący ruch gałek podczas śledzenia poruszającego się obiektu. Wart podkreślenia jest fakt, iż poza czynnościami motorycznymi aktywna wizja obejmuje wykonywanie wszystkiech czynności ułatwiających percepcję, a więc również takie działania jak zmiana parametrów kamery (np. ostrość czy zbliżenie), zmiana sposobu percepcji (algorytmu analizy i rozpoznawania obrazów) lub aktywny wpływ na oświetlenie sceny – przykładowo, w pracy [3] zaproponowano wykorzystanie dodatkowego źródła światła, którego zarówno położenie, jak i intensywność, są kontrolowane przez system sterujący.

W tej pracy omówiono system sterowania robota-kasjera, którego zadaniem jest sterowanie manipulatorem wyposażonym w kamerę zintegrowaną z jego chwytakiem (rys. 1) w celu odczytania kodów kreskowych produktów znajdujących się na kontuarze. Dla ułatwienia przyjęto założenie, iż samym wyłożeniem produktów, jak i odwróceniem ich kodów w odpowiednią stronę, zajmie się klient. Dlatego przy specyfikacji systemu sterowania uwagę skupiono

Rys. 1. Manipulator IRp-6 w trakcie identyfikacji produktów
Fig. 1. The IRp-6 manipulator during products identification

na samej aktywnej percepcji wizyjnej, a system zdekomponowano na zestaw różnorodnych zachowań, od ruchu pozycyjnego, przez przeglądanie lady w celu znalezienia obszaru podobnego do kodu i odpowiednie przybliżenie się do tego obszaru, aż po samą próbę odczytu kodu.

System sterowania robota-kasjera wyspecyfikowano za pomocą formalnego opisu systemów robotycznych, zaprezentowanego w pracy [4]. Formalizm ten wykorzystuje dwie fundamentalne koncepcje zaczerpnięte z informatyki oraz teorii automatów: podejście agentowe oraz opis działania elementów poprzez funkcje przejścia.

1. Wykorzystane narzędzia

Programowa struktura ramowa **MRROC++** (ang. *Multi–Robot Research Oriented Controller*) [11] jest narzędziem do tworzenia hierarchicznych, rozproszonych sterowników systemów wielorobotowych. Została ona z powodzeniem wykorzystana w różnorodnych aplikacjach robotycznych, m.in. do powielania rysunków, chwytania obiektów oraz

układania kostki Rubika [10]. W skład sterownika systemu robotycznego opartego o programową strukturę ramową **MRROC++** wchodzą podzielone na warstwy procesy (rys. 2):

- UI – pojedynczy proces interfejsu użytkownika odpowiedzialny za komunikację z operatorem,
- MP – pojedynczy proces koordynujący pracę wszystkich efektorów,
- ECP – proces odpowiedzialnych za realizację zadania zleconego pojedynczemu efektorowi,
- EDP – proces odpowiedzialny za bezpośrednie sterowanie pojedynczym efektorem,
- VSP – proces odpowiedzialny za agregację danych sensorycznych do postaci użytecznej w sterowaniu. W zależności od liczby czujników i rodzaju sprzężenia, procesy VSP mogą być skojarzone z ECP lub MP.

Ponieważ omawiany system sterujący robota-kasjera jest systemem jednorobotowym, dlatego proces koordynujący (MP) nie jest istotny, a uwaga została skupiona na agencie sterującym manipulatorem, nazwanym agentem a_1. Na jego układ sterowania składają się procesy: EDP stanowiący niskopoziomowy sterownik manipulatora, VSP odpowiedzialny za agregację informacji wizyjnej pochodzącej z kamery oraz pełniący naczelną rolę ECP, odpowiedzialny za realizację całego zadania (rys. 4).

Wizyjna struktura ramowa **FraDIA** (ang. *Framework for Digital Image Analysis*) [7] powstała w odpowiedzi na zapotrzebowanie na narzędzie ułatwiające oraz standaryzujące proces tworzenia aplikacji wizyjnych. Zaimplementowane w jej warstwie sprzętowej sterowniki źródeł obrazu pozwalają na skupienie uwagi programisty na właściwych algorytmach analizy oraz rozpoznawania obrazów. Ponieważ zrąb oferuje dodatkowo gotowe mechanizmy do komunikacji z procesami systemu **MRROC++**, obie struktury mogą być użyte do konstrukcji złożonych sterowników robotycznych wykorzystujących wizję komputerową. W omawianym systemie zadanie wizyjne działające w ramach procesu **FraDIA** pełni rolę wyspecjalizowanego procesu VSP systemu **MRROC++**. Należy zauważyć, iż **FraDIA** należy jednocześnie zarówno do warstwy sprzętowej, jak i zadaniowej, gdyż z jednej strony jest sterownikiem kamery, a z drugiej realizuje zadanie wizyjne, czyli konkretny algorytm przetwarzania i analizy obrazu. Ponieważ wymagane były różnorodne procesy percepcji wizyjnej, dlatego w ramach **FraDIA** stworzone zostały w istocie trzy różne zadania wizyjne, przełączane na żądanie **MRROC++**.

2. Przyjęta notacja

Zachowanie oznaczone zostało jako ${}^{m}\mathcal{B}_i$, gdzie m jest numerem zachowania i-tego agenta. Każde zachowanie ${}^{m}\mathcal{B}_i\left({}^{m}f_{c_{c_i}}, {}^{m}f_{c_{e_i}}, {}^{m}f_{c_{V_i}}, {}^{m}f_{c_{T_i}}, {}^{m}f_{\tau_i}\right)$ zdefiniowano przez pięć funkcji, z których każda korzysta z tych samych argumentów: c_{c_i} to pamięć agenta, przez bufor wejściowy ${}_{x}c_{e_i}$ odbierany jest aktualny stan efektora, w buforze wejściowym ${}_{x}c_{V_i}$ przechowywany jest aktualny odczyt z czujnika wirtualnego, natomiast ${}_{x}c_{T_i}$ jest buforem służącym do odbioru informacji przesyłanych przez inne agenty. Funkcja ${}^{m}f_{c_{c_i}}$ zmienia stan pamięci agenta, ${}^{m}f_{c_{e_i}}$ oblicza sterowanie wysyłane do efektora przez bufor wyjściowy ${}_{y}c_{e_i}$, ${}^{m}f_{c_{V_i}}$ służy do obliczania rozkazu dla czujnika wirtualnego, zapisywanego do bufora wyjściowego ${}_{y}c_{V_i}$, a ${}^{m}f_{c_{T_i}}$ tworzy wiadomość wysyłaną do innych agentów przez bufor ${}_{y}c_{T_i}$. Funkcja ${}^{m}f_{\tau_i}$ jest warunkiem terminalnym działania zachowania.

W artykule przyjęto następującą konwencję opisu zmiennych ${}^{X}_{Y}\mathcal{Z}_{W}$: $\mathcal{Z} \in \{\mathcal{T}, \mathcal{O}, \mathcal{F}, \Delta\}$ jest głównym symbolem, gdzie $\mathcal{T}$ oznacza transformacje pomiędzy układami współrzędnych, $\mathcal{O}$ to obiekt, $\mathcal{F}$ – cecha, natomiast Δ oznacza przyrost położenia. $X \in \{0, E, C\}$ opisuje układ współrzędnych: 0 – układ współrzędnych związany z podstawą robota (globalny), E – układ współrzędnych związany z końcówką roboczą, C – układ współrzędnych związany z kamerą. Y może być związany z danym obiektem lub cechą (np. l-ty obiekt) lub układem współrzędnych (np. ${}^{0}_{E}\mathcal{T}$ oznacza położenie układu efektora robota względem układu globalnego). Symbol $W \in \{c, d, p, u\}$ jest związany z wartością zmiennej: c to wartość aktualna, p to wartość poprzednia, u to nowo obliczona wartość, natomiast d jest wartością pożądaną. Symbolem $\bullet$ oznaczono argumenty lub zmienne nieistotne (nie brane pod uwagę w danej funkcji).

3. Specyfikacja agenta a_1

Diagram stanów agenta obrazujący przełączanie się pomiędzy zachowaniami pokazany został na rys. 3. Agent a_1 jest odpowiedzialny za realizację czterech zachowań elementarnych:

- ${}^{1}\mathcal{B}_1$ – przegląd lady w celu znalezienia obszarów występowania obiektów podobnych do kodów kreskowych,
- ${}^{2}\mathcal{B}_1$ – ruch manipulatora mający na celu odpowiednie ustawienie kamery nad obszarem podobnym do kodu kreskowego,
- ${}^{3}\mathcal{B}_1$ – identyfikacja, czyli odczytanie kodu kreskowego,
- ${}^{4}\mathcal{B}_1$ – ruch do zadanej pozycji (np. ruch do pozycji startowej lub powrót do pozycji, w której zachowanie ${}^{1}\mathcal{B}_1$ zostało przełączone na zachowanie ${}^{2}\mathcal{B}_1$).

Ponieważ zachowania ${}^{1}\mathcal{B}_1$, ${}^{2}\mathcal{B}_1$ oraz ${}^{3}\mathcal{B}_1$ wymagały różnych funkcji agregujących dane sensoryczne, dlatego wraz ze zmianą zachowania agenta a_1 przełączane jest również zadanie wizyjne. Zadanie wizyjne ${}^{i}V_1$ zostało związane z zachowaniem ${}^{i}\mathcal{B}_1$, gdzie $i = 1, 2, 3$. W zachowaniu ${}^{4}\mathcal{B}_1$ informacja wizyjna nie jest wykorzystywana – jest to czysty ruch pozycyjny.

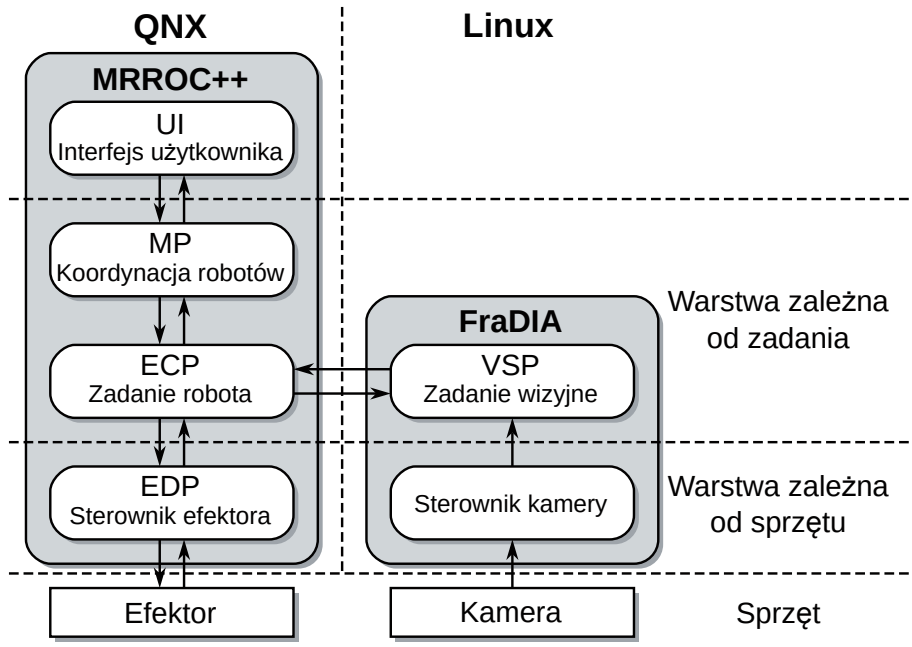
3.1. Struktura obrazów agenta a_1

Struktura agenta a_1, sterującego zmodyfikowanym manipulatorem IRp-6 [9] wyposażonym w kamerę zintegrowaną z chwytakiem, została pokazana na rys. 4.

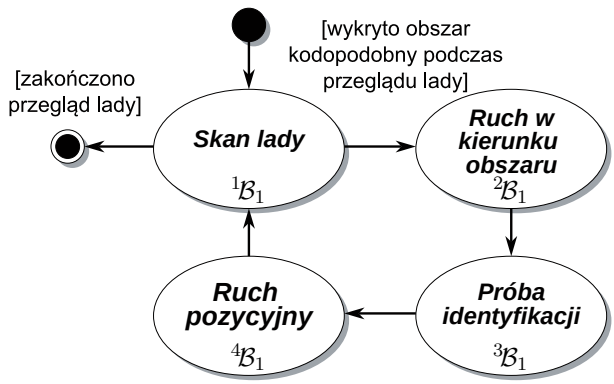
3.1.1. Pamięć wewnętrzna

Agent a_1 przechowuje w pamięci c_{c_1} następujące dane:

$$
\begin{aligned}
&c_{c_1}\left[{}^{E}_{C}\mathcal{T}\right] && \text{– transformację pomiędzy chwytakiem a kamerą,}\\
&c_{c_1}\left[{}^{0}_{s}\mathcal{T}\right] && \text{– pozycję, z której przeglądanie lady ma się rozpoczynać,}\\
&c_{c_1}\left[{}^{0}_{e}\mathcal{T}\right] && \text{– pozycję, w której przeglądanie lady ma się zakończyć,}\\
&c_{c_1}\left[{}^{0}\mathcal{O}\right] && \text{– listę wykrytych obszarów kodopodobnych.}
\end{aligned}
\tag{1}
$$



Rys. 2. Struktura sterowników ze sprzężeniem wizyjnym opartych na **MRROC++** i **FraDIA**
Fig. 2. General structure of robotic controllers with visual feedback, based on the **MRROC++** and **FraDIA** frameworks



Rys. 3. Diagram stanów systemu sterowania agenta a_1
Fig. 3. State diagram of the agent a_1 control system

Na liście $c_{c_1}\left[{}^0\mathcal{O}\right]$ przechowywane są wszystkie wykryte obszary kodopodobne:

$$ {}^0\mathcal{O} \;=\; \left[{}^0_1\mathcal{O}\,,\,{}^0_2\mathcal{O}\,,\,\ldots\right] \tag{2}$$

z których każdy l-ty obszar opisany jest wektorem trzech cech:

$$ {}^0_l\mathcal{O} \;=\; \left[{}^0_l\mathcal{F}\right] \;=\; \left[{}^0_l\mathcal{T}\,,\,{}_l\texttt{size}\,,\,{}_l\texttt{code}\right] \tag{3}$$

gdzie ${}^0_l\mathcal{T}$ jest pozycją tego obszaru w globalnym układzie odniesienia, ${}_l\texttt{size}$ przechowuje jego wielkość w obrazie, natomiast w ${}_l\texttt{code}$ zapamiętywany jest odczytany kod kreskowy (lub -2 w przypadku niepoprawnego kodu). Wartość -1 jest wartością domyślną wpisywaną w pole ${}_l\texttt{code}$ nowego obszaru kodopodobnego, natomiast poprawnie odczytane kody przyjmują wartości większe bądź równe 0.

3.1.2. Obrazy czujników wirtualnych

Czujnik wirtualny komunikuje się z układem sterowania agenta poprzez bufory ${}^m_x c_V$ oraz ${}^m_y c_V$ zwane obrazami. Zadanie wizyjne ${}^1V_1$ jest związane z zachowaniem ${}^1\mathcal{B}_1$, a więc z lokalizacją obszarów zawierających obiekty podobne do kodów kreskowych. Do czujnika nie są przesyłane żadne dane, a więc bufor ${}^1_y c_{V_1} = [\bullet]$. Wejściowy obraz czujnika wirtualnego ${}^1_x c_{V_1}$ zawiera listę wykrytych obszarów, których pozycje wyrażono w układzie kamery:

$$ {}^1_x c_{V_1}\left[{}^C\mathcal{O}\right] - \text{obszary kodopodobne wykryte w danym obrazie.} \tag{4}$$

Każdy z obiektów opisany jest wektorem cech (3), natomiast warte podkreślenia jest, iż zadanie ${}^1V_1$ zwraca jedynie pozycję oraz rozmiar wykrytych obszarów, a więc:

$$ {}^C_l\mathcal{O} \;=\; \left[{}^C_l\mathcal{F}\right] \;=\; \left[{}^C_l\mathcal{T}\,,\,{}_l\texttt{size}\right] \tag{5}$$

Zachowanie ${}^2\mathcal{B}_1$ realizuje ruch manipulatora w kierunku obszaru, który potencjalnie może zawierać kod, dlatego informacja o pozycji obszaru, do którego będzie wykonywany ruch, musi zostać wysłana do VSP poprzez obraz wyjściowy czujnika:

$$ {}^2_y c_{V_1}\left[{}^C_l\mathcal{T}\right] - \text{pozycja obszaru w obrazie.} \tag{6}$$

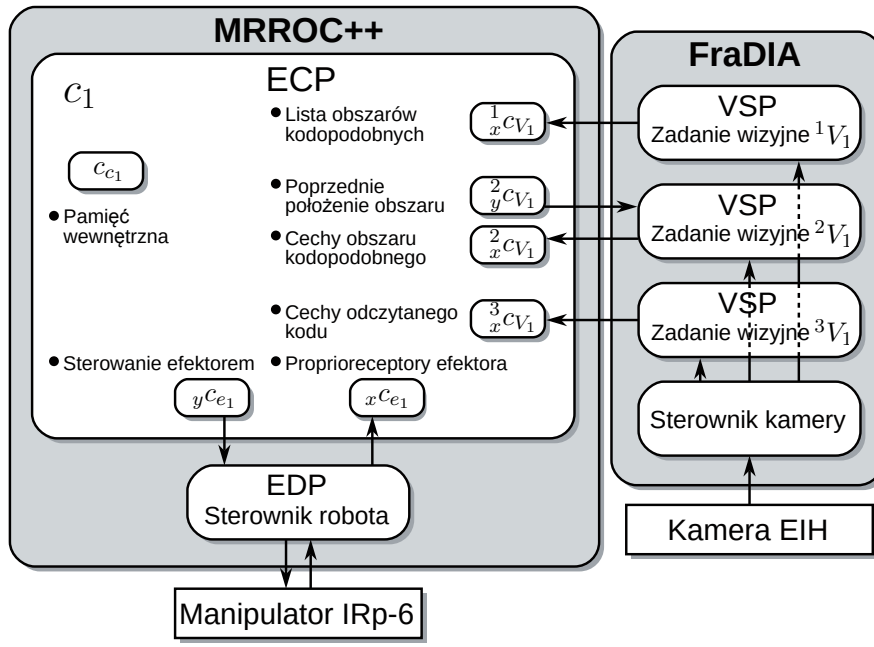
Obraz wejściowy czujnika wirtualnego ${}^2_x c_{V_1}$ związanego z zadaniem ${}^2V_1$ zawiera:

$$ {}^2_x c_{V_1}\left[{}^C_l\mathcal{F}\right] \;=\; \left[{}^C_l\mathcal{T}\,,\,{}_l\texttt{size}\right] - \text{cechy obszaru kodopodobnego.} \tag{7}$$

Zachowanie ${}^3\mathcal{B}_1$ odpowiedzialne jest za identyfikację kodu kreskowego, stąd obraz wejściowy czujnika wirtualnego ${}^3_x c_{V_1}$ związanego z zadaniem ${}^3V_1$ ma następującą postać:

$$ {}^3_x c_{V_1}\left[{}^C_l\mathcal{F}\right] \;=\; \left[{}^C_l\mathcal{T}\,,\,{}_l\texttt{code}\right] - \text{cechy zidentyfikowanego obszaru.} \tag{8}$$

Bufor wyjściowy nie jest używany ${}^3_y c_{V_1} = [\bullet]$.



Rys. 4. Struktura systemu sterowania agenta a_1
Fig. 4. Structure of the agent a_1 control system

3.1.3. Obrazy efektora

Wyjściowy bufor efektora $_y c_{e_1}$ zawiera przyrost położenia efektora, który ma zostać wykonany:

$$_y c_{e_1} \left[{}^E\Delta \right] \text{ – przyrost położenia efektora,} \tag{9}$$

natomiast bufor wejściowy $_x c_{e_1}$ zawiera aktualną pozycję końcówki robota względem układu globalnego:

$$_x c_{e_1} \left[{}^0_E\mathcal{T} \right] \text{ – aktualna pozycja efektora.} \tag{10}$$

3.1.4. Obrazy transmisyjne

Agent a_1 nie kontaktuje się z innymi agentami, stąd:

$$_x c_{T_1} \;=\; _y c_{T_1} \;=\; [\bullet] \tag{11}$$

3.2. Zachowanie $^1\mathcal{B}_1$: *Skan lady*

Lokalizacja obszarów występowania obiektów podobnych do kodów kreskowych jest realizowane przez zachowanie $^1\mathcal{B}_1$. Ruch wykonywany jest po prostej w przestrzeni kartezjańskiej łączącej punkty startowy $^0_s\mathcal{T}$ oraz końcowy $^0_e\mathcal{T}$. W trakcie jego wykonywania badany jest fragment przestrzeni, który obserwuje kamera (skierowana na ladę). Ważne przy tym jest, iż:

– wykorzystywane zadanie wizyjne $^1V_1$ w każdym kroku może wykryć wiele obszarów kodopodobnych,

– w każdym kroku zlokalizowane obszary kodopodobne zapamiętywane są w pamięci agenta a_1.

Założenia te implikują, iż system na podstawie aktualnych danych sensorycznych, pozycji chwytaka i kamery musi rozpoznać, które z aktualnie zaobserwowanych obszarów znajdują się już w pamięci, a które są nowymi – realizuje to funkcja przejścia $^1f_{c_{c_1}}$. Funkcja przejścia $^1f_{c_{e_1}}$ odpowiada za doprowadzenie efektora do pozycji docelowej. Funkcje $^1f_{c_{T_1}}$ oraz $^1f_{c_{V_1}}$ nie są wykorzystane, a więc argumenty tego zachowania są następujące: $^1\mathcal{B}_1\left(^1f_{c_{c_1}}, ^1f_{c_{e_1}}, \bullet, \bullet, ^1f_{\tau_1}\right)$.

Funkcja przejścia $^1f_{c_{e_1}}$: W omawianym zachowaniu wykonywany jest ruch do z góry zadanej pozycji, niezależnie od tego, co aktualnie jest obserwowane. Funkcja $^1f_{c_{e_1}}$ została opisana za pomocą przekształceń zdefiniowanych wzorami (12-13), a jej najważniejsze bloki pokazane zostały na rys. 5. Funkcja obliczająca uchyb zależy jedynie od docelowej oraz aktualnej pozycji efektora:

$$^E\mathcal{T}_{e,c} \;=\; {}^{E_c}_{e}\mathcal{T} \;=\; \left({}^0_E\mathcal{T}_c\right)^{-1} \; {}^0_e\mathcal{T} \tag{12}$$

Na jego podstawie regulator oblicza sterowanie, będące w istocie przyrostem położenia efektora, stanowiącym mały fragment uchybu (12). Przyrost ten zostanie zrealizowany w ciągu kolejnego kroku sterowania:

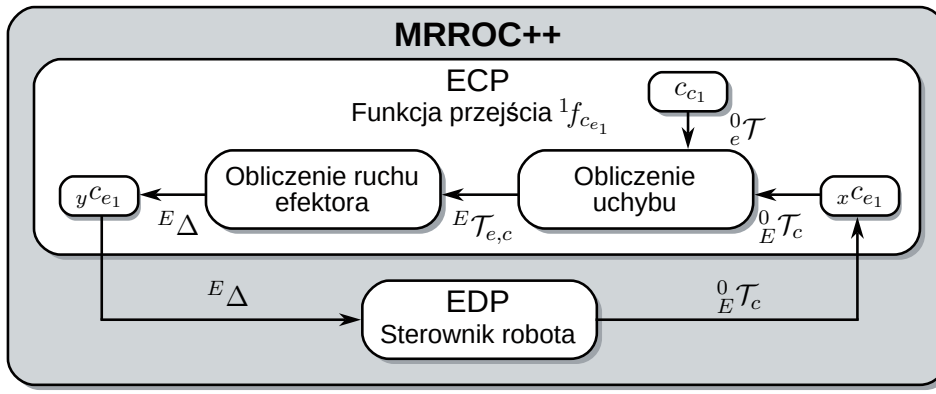
$$^E\Delta = \texttt{generatestep}\left(^E\mathcal{T}_{e,c}\right) \tag{13}$$

W tym celu obliczony przyrost jest wysyłany do sterownika manipulatora – procesu EDP, który bezpośrednio steruje robotem.

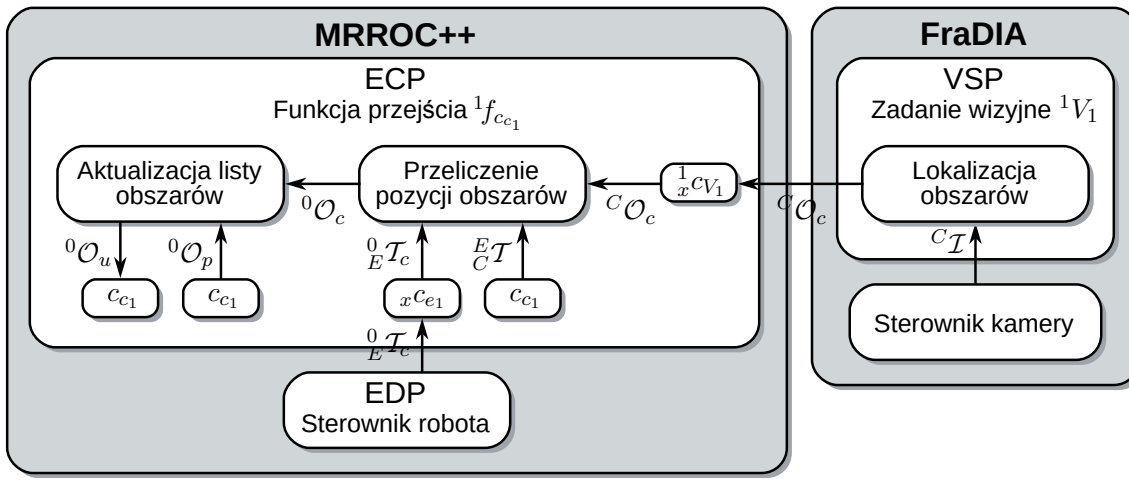
Funkcja przejścia $^1f_{c_{c_1}}$: Najważniejsze bloki związane z funkcją $^1f_{c_{c_1}}$ pokazane zostały na rys. 6. Z zadania wizyjnego $^1V_1$ pozyskiwana jest informacja dotycząca wszystkich l_n obszarów kodopodobnych $^C\mathcal{O}_c$ wykrytych w aktualnym obrazie $^C\mathcal{I}$, przy czym każdy z obiektów na liście posiada odpowiedni wektor cech (5). W pierwszym kroku pozycje wszystkich l_n zlokalizowanych obszarów, na podstawie znajomości aktualnej pozycji efektora $^0_E\mathcal{T}_c$ oraz transformacji pomiędzy efektorem a kamerą $^E_C\mathcal{T}$, zostają przeliczone do globalnego układu odniesienia:

$$^0_l\mathcal{T}_c \;=\; {}^0_E\mathcal{T}_c \; {}^E_C\mathcal{T} \; {}^C_l\mathcal{T}_c, \forall l = 1, \ldots, l_n \tag{14}$$

Położenie każdego (l-tego) obszaru wraz z pozostałymi cechami zapamiętywane są w $^0_l\mathcal{O}_c$. Tak przetworzone dane aktualnie obserwowanych obszarów $^0\mathcal{O}_c$ wykorzystane są



Rys. 5. Diagram przepływu danych funkcji przejścia ${}^1f_{c_{e_1}}$
Fig. 5. Data flow diagram of the ${}^1f_{c_{e_1}}$ transfer function



Rys. 6. Diagram przepływu danych funkcji przejścia ${}^1f_{c_{c_1}}$
Fig. 6. Data flow diagram of the ${}^1f_{c_{c_1}}$ transfer function

następnie do aktualizacji przechowywanej w pamięci agenta listy obszarów ${}^0\mathcal{O}_u$:

$$ {}^0\mathcal{O}_u \;=\; \texttt{update}\left({}^C\mathcal{O}_c\,,\,{}^C\mathcal{O}_p\right) \tag{15}$$

Warunek końcowy ${}^1f_{\tau_1}$: Warunek końcowy obejmuje zakończenie wykonywania zachowania poprzez zakończenie ruchu (osiągnięcie pozycji, w której przeglądanie lady jest zakończone) lub na przechowywanej w pamięci liście obszarów pojawi się nowy, niezbadany jeszcze obszar:

$$ {}^1f_{\tau_1} \;\triangleq\; \left({}^0_E\mathcal{T}_c \;=\; {}^0_e\mathcal{T}\right) \vee \left(\texttt{unidentifiedareaexists}\left({}^0\mathcal{O}_u\right)\right) \tag{16}$$

3.3. Zachowanie ${}^2\mathcal{B}_1$: *Ruch nad obszar*

Zachowanie przybliżania się do obszaru kodopodobnego ${}^2\mathcal{B}_1$ realizowane jest przez serwomechanizm wizyjny operujący na uchybie w przestrzeni cech obrazu. Porównywane cechy obszaru to położenie jego środka (pożądane jest, aby obszar znajdował się w środku obrazu) oraz jego rozmiar (pożądany jest odpowiednio duży).

Zachowanie to wykorzystuje zadanie wizyjne ${}^2V_1$, którego celem jest wyznaczanie położenia oraz wielkości obszaru kodopodobnego. Zadanie to różni się od ${}^1V_1$ tym, iż skupia się jedynie na jednym wybranym obiekcie, natomiast ${}^1V_1$ jednocześnie wyznacza położenia wszystkich widocznych obszarów kodopodobnych.

Argumenty zachowania ${}^2\mathcal{B}_1$ zostały zdefiniowane jako ${}^2\mathcal{B}_1\left(\bullet, {}^2f_{c_{e_1}}, {}^2f_{c_{V_1}}, \bullet, {}^2f_{\tau_1}\right)$, a więc funkcje przejścia ${}^2f_{c_{c_1}}$ oraz ${}^2f_{c_{T_1}}$ nie są wykorzystywane.

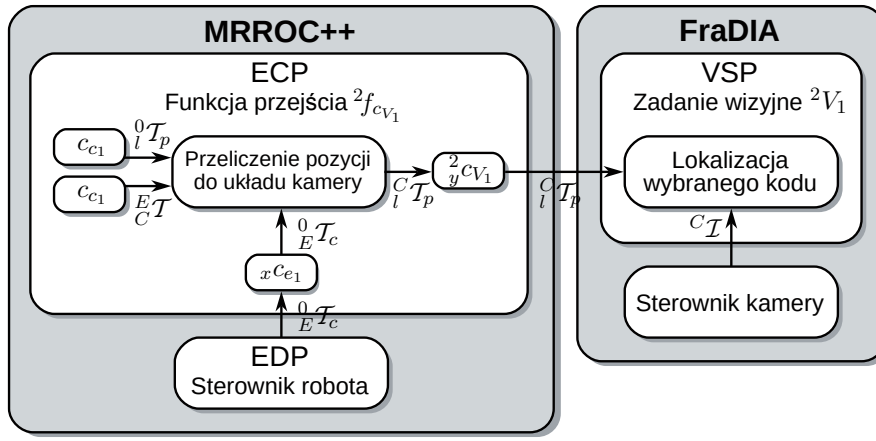
Funkcja przejścia ${}^2f_{c_{V_1}}$: Na rys. 7 pokazane zostały najważniejsze bloki funkcji. Funkcja ta wysyła do zadania wirtualnego poprzednie położenie l-tego środka obszaru ${}^C_l\mathcal{T}_p$, obliczone na podstawie położenia tego obszaru w układzie bazowym robota, aktualnego położenia efektora oraz transformacji pomiędzy kamerą a efektorem:

$$ {}^C_l\mathcal{T}_p \;=\; \left({}^0_E\mathcal{T}_c\;{}^E_C\mathcal{T}\right)^{-1}\;{}^0_l\mathcal{T}_p \tag{17}$$

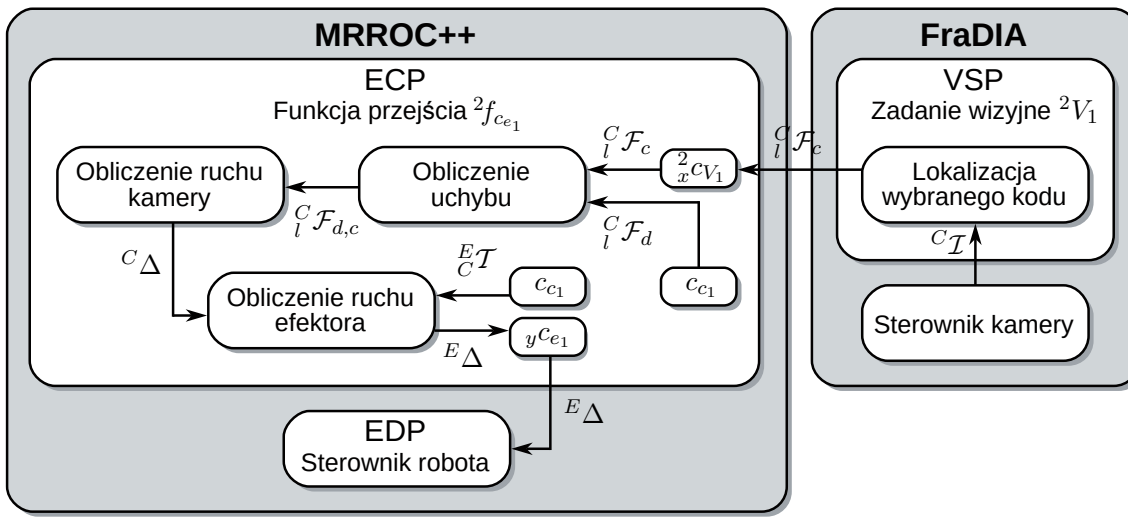
Funkcja przejścia ${}^2f_{c_{e_1}}$: Najważniejsze bloki funkcji pokazane zostały na rys. 8, a jej działanie jest opisane wzorami (18–20). Na podstawie porównania obserwowanych ${}^C_l\mathcal{F}_c$ oraz pożądanych ${}^C_l\mathcal{F}_d$ cech l-tego obszaru wyznaczany jest uchyb w przestrzeni cech obrazu:

$$ {}^C_l\mathcal{F}_{d,c} \;=\; {}^C_l\mathcal{F}_d \;-\; {}^C_l\mathcal{F}_c \tag{18}$$

Na podstawie uchybu w przestrzeni cech obliczany jest ruch kamery – przyrost położenia, które zostanie osiągnięte



Rys. 7. Diagram przepływu danych funkcji przejścia ${}^2f_{c_{V_1}}$
Fig. 7. Data flow diagram of the ${}^2f_{c_{V_1}}$ transfer function



Rys. 8. Diagram przepływu danych funkcji przejścia ${}^2f_{c_{e_1}}$
Fig. 8. Data flow diagram of the ${}^2f_{c_{e_1}}$ transfer function

w ciągu kolejnego kroku sterowania:

$$ {}^C\Delta = \texttt{generatestep}\left({}^C\mathcal{F}_{d,c}\right) \tag{19}$$

który zostaje przekształcony do przyrostu położenia efektora:

$$ {}^E\Delta \;=\; {}^E_C\mathcal{T}\;{}^C\Delta \tag{20}$$

Warunek końcowy ${}^2f_{\tau_1}$: Zachowanie ${}^2\mathcal{B}_1$ kończy swoje działanie, gdy obserwowane cechy obszaru kodopodobnego będą odpowiednie, aby podjąć próbę jego odczytania, a więc gdy odpowiednie będą jego położenie oraz rozmiar:

$$ {}^2f_{\tau_1} \;\triangleq\; \left(\texttt{adequate}({}^C_l\mathcal{F}_c)\right) \tag{21}$$

3.4. Zachowanie ${}^3\mathcal{B}_1$: *Próba identyfikacji*

Zachowanie ${}^3\mathcal{B}_1$ jest pasywne, tzn. nie jest wykonywany żaden ruch. Zadaniem tego zachowania jest zapamiętanie wyniku próby odczytu kodu kreskowego, odebranego od zadania wizyjnego ${}^3V_1$. Warunkiem zakończenia jest odebranie odczytu – zidentyfikowanego kodu lub też informacji o tym, iż obszar nie zawiera kodu.

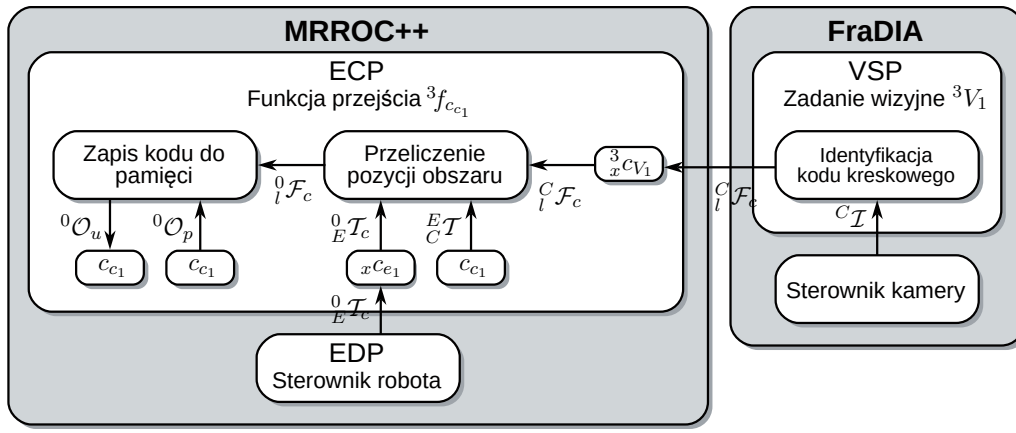
Funkcje przejścia związane z ruchem efektora ${}^3f_{c_{e_1}}$, rozkazami wysyłanymi do VSP ${}^3f_{c_{V_1}}$ oraz komunikacją z innymi agentami ${}^3f_{c_{T_1}}$ są zbędne, a więc argumenty zachowanie mają postać ${}^3\mathcal{B}_1\left({}^3f_{c_{c_1}}, \bullet, \bullet, \bullet, {}^3f_{\tau_1}\right)$.

Funkcja przejścia ${}^3f_{c_1}$: Głównym celem tej funkcji jest odpowiednie przetworzenie odebranej informacji sensorycznej. Na rys. 9 pokazane zostały najważniejsze jej bloki. W pierwszym kroku położenie obszaru w obrazie, stanowiące jeden z elementów opisującego go wektora cech (8), jest przeliczane do układu bazowego:

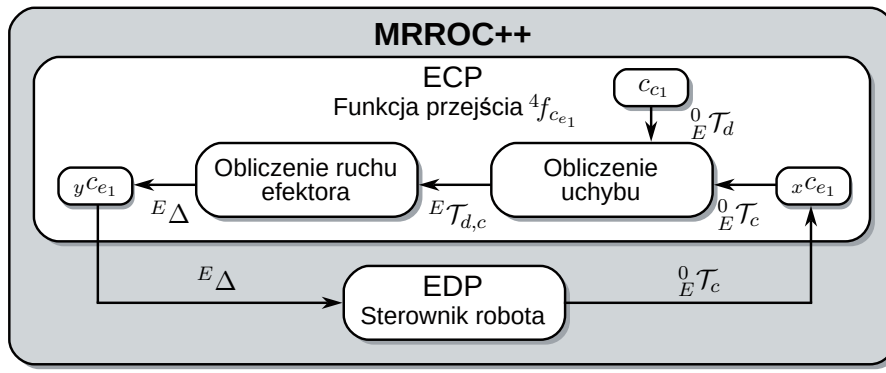
$$ {}^0_l\mathcal{T}_c \;=\; {}^0_E\mathcal{T}_c\;{}^E_C\mathcal{T}\;{}^C_l\mathcal{T}_c \tag{22}$$

Następnie cechy obszaru wykorzystywane są do aktualizacji istniejącej w pamięci listy zlokalizowanych obszarów kodobpodobnych:

$$ {}^0\mathcal{O}_u \;=\; \texttt{update}\left({}^0_l\mathcal{F}_c\,,\,{}^C\mathcal{O}_p\right) \tag{23}$$



Rys. 9. Diagram przepływu danych funkcji przejścia ${}^3f_{c_{c_1}}$
Fig. 9. Data flow diagram of the ${}^3f_{c_{c_1}}$ transfer function



Rys. 10. Diagram przepływu danych funkcji przejścia ${}^4f_{c_{e_1}}$
Fig. 10. Data flow diagram of the ${}^4f_{c_{e_1}}$ transfer function

Warunek końcowy ${}^3f_{\tau_1}$: Warunek końcowy związany jest z wykonaniem aktualizacji listy obszarów:

$$ {}^3f_{\tau_1} \triangleq \left(\text{updated}\left({}^0\mathcal{O}_u\right)\right) \tag{24}$$

3.5. Zachowanie ${}^4\mathcal{B}_1$: *Ruch do pozycji*

Zachowanie ${}^4\mathcal{B}_1$ jest związane z ruchem do zadanej pozycji bez wykorzystania informacji sensorycznych, dlatego funkcje ${}^4f_{c_{c_1}}$, ${}^4f_{c_{\mathcal{T}_1}}$ oraz ${}^4f_{c_{V_1}}$ nie są wykorzystywane, a więc ${}^4\mathcal{B}_1\left(\bullet, {}^4f_{c_{e_1}}, \bullet, \bullet, {}^4f_{\tau_1}\right)$.

Funkcja przejścia ${}^4f_{c_{e_1}}$: Najważniejsze bloki pokazane zostały na rys. 10. Działanie tej funkcji jest podobne do funkcji ${}^1f_{c_{e_1}}$, która została opisana za pomocą przekształceń zdefiniowanych wzorami (12–13). Różnica polega jedynie w docelowej pozycji – zamiast położenia końcowego ${}^0_e\mathcal{T}$ zadawane mogą być różne położenia docelowe ${}^0_E\mathcal{T}_d$, np. pozycja, w której zachowanie ${}^1\mathcal{B}_1$ zostało przerwane z powodu wykrycia nowego obszaru kodopodobnego.

Warunek końcowy ${}^4f_{\tau_1}$: Warunek końcowy związany jest z osiągnięciem zadanej pozycji docelowej:

$$ {}^4f_{\tau_1} \triangleq \left({}^0_E\mathcal{T}_c = {}^0_E\mathcal{T}_d\right) \tag{25}$$

4. Podsumowanie

Projekt złożonego systemu sterowania wymaga jego dekompozycji na bloki łatwe w implementacji oraz łączeniu. W artykule przedstawiono metodę dekompozycji systemu sterowania pojedynczego agenta na zachowania odpowiedzialne za realizację różnorodnych podzadań. Wyspecyfikowane zachowania posłużyły jako baza do stworzenia systemu, w którym do sterowania robotem wykorzystano programową strukturę ramową **MRROC++**, natomiast percepcja wizyjna zaimplementowana została za pomocą wizyjnej struktury ramowej **FraDIA**. Wart jednak podkreślenia jest fakt, iż zaprezentowane formalne podejście może zostać z powodzeniem zastosowne w systemach bazujących na innych robotycznych strukturach ramowych.

Wybrane do specyfikacji zadanie stanowi ciekawy przykład zastosowania aktywnej wizji, zawierający różnorodne aktywne zachowania: od zmiany samego sposobu percepcji, przez przeglądanie sceny, aż po zbliżanie się do wybranych obszarów w celu ich dokładniejszej analizy. Omawiany system wykazuje, iż aby robot w rzeczywistości wykorzystywał aktywne czucie, niezbędna jest kombinacja nie tylko aktywnych i pasywnych zachowań, ale również zachowań wykorzystujących zmysły oraz nie korzystających z nich.

Wyspecyfikowane zachowania zostały zaimplementowane w oparciu o struktury ramowe **MRROC++** oraz **FraDIA**, a ich działanie zweryfikowano podczas identyfikacji rzeczywistych obiektów przez robota w postaci zmodyfikowanego manipulatora IRp-6 wyposażonego w kamerę zintegrowaną z jego chwytakiem (rys. 1). Obecnie trwają prace ich

integracją w system realizujący w pełni autonomiczną identyfikację obiektów położonych na ladzie.

Docelowo w systemie robota-kasjera planowane jest wykorzystanie dwóch manipulatorów IRp-6 oraz robota w postaci taśmociągu, których współpraca znacznie przyśpieszy działanie systemu. System sterowania będzie złożony wtedy z trzech agentów upostaciowionych (z których każdy odpowiedzialny będzie za sterowanie jednym robotem) oraz agenta koordynującego, odpowiedzialnego za zlecanie wykonywanie poszczególnych zadań innym agentom. Należy tutaj zaznaczyć, iż w systemie tym będą mogły zostać z powodzeniem wykorzystane wyspecyfikowane w artykule zachowania, rozbudowane dodatkowo o funkcje ${}^{m}f_{c_{T_i}}$, związane z komunikacją z koordynatorem.

Podziękowania

Praca finansowana przez grant Ministerstwa Nauki i Szkolnictwa Wyższego N514 1287 33.

Bibliografia

1. J. Aloimonos, I. Weiss, A. Bandyopadhyay. Active vision. *Proceedings of 1st International Conference on Computer Vision*, strony 35–54, 1987.
2. Ruzena Bajcsy, Mario Campos. Active and exploratory perception. *Technical Report (CIS)*, 1991.
3. S.Y. Chen, Jianwei Zhang, Houxiang Zhang, Wanliang Wang, Y.F. Li. Active illumination for robot vision. *IEEE International Conference on Robotics and Automation*, 2007.
4. C. Zieliński. Formalne podejście do programowania robotów – struktura układu sterującego, rozdział w *Inteligencja wokół nas. Współdziałanie agentów softwareowych, robotów, inteligentnych urządzeń*, strony 267–300. EXIT, 2010.
5. D. Floreano, M. Suzuki. Active vision and neural development in animals and robots. *Proceedings of the Seventh International Conference on Cognitive Modeling*, strony 10–11, 2006.
6. Youngmo Han. Imitation of human-eye motion – how to fix gaze of an active vision system. *IEEE Transactions on Systems, Man, and Cybernetics*, wolumen 37, 2007.
7. Tomasz Kornuta. Application of the FraDIA vision framework for robotic purposes. *Proceedings of the International Conference on Computer Vision and Graphics, Part II*, wolumen 6375 serii *Lecture Notes in Computer Science*, strony 65–72. Springer, Berlin, Heidelberg, 2010.
8. T. Maruszewski. *Psychologia poznania.* Gdańskie Wydawnictwo Psychologiczne, Gdańsk, 2003.
9. C. Zieliński, W. Szynkiewicz, K. Mianowski, A. Rydzewski, T.Winiarski. Efektory robota usługowego do dwuręcznej manipulacji z czuciem. K. Tchoń, redaktor, *IX Krajowa Konferencja Robotyki – Postępy Robotyki: Systemy i współdziałanie robotów*, wolumen 2, strony 257–266. Wydawnictwa Komunikacji i Łączności, Warszawa, 2006.
10. C. Zieliński, W. Szynkiewicz, T. Winiarski, M. Staniak, W. Czajewski, T. Kornuta. Rubik's cube as a benchmark validating MRROC++ as an implementation tool for service robot control systems. *Industrial Robot: An International Journal*, 34(5):368–375, 2007.
11. C. Zieliński, T. Winiarski. Motion Generation in the MRROC++ Robot Programming Framework. *The International Journal of Robotics Research*, 29(4):386–413, 2010.

Decomposition of the robot-cashier control system

Abstract: The paper presents a formal method of specification of control systems on the example of a robot cashier. The idea of this controller was based on the active vision paradigm. Its aim is to analyze selected scene fragments, in this case in order to identify the object by reading its barcode. The specification contains a set of diverse, but simple, behaviors which when integrated create a controller capable of realizing the robot's goal.

Keywords: robot cashier, control system, behavior, active vision

mgr inż. Tomasz Kornuta

Absolwent Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej. W 2003 roku uzyskał tytuł inżyniera, w 2005 tytuł magistra inżyniera. Od 2008 roku pracuje na etacie asystenta w Instytucie Automatyki i Informatyki Stosowanej (IAiIS), w ramach którego prowadzi zajęcia dydaktyczne. Dodatkowo od 2009 roku jest zatrudniony na etacie konstruktora, w ramach którego realizuje prace związane z grantem europejskim. Od 2005 roku w ramach doktoratu prowadzi badania związane z wykorzystaniem przez roboty paradygmatu aktywnego czucia do analizy otoczenia. Jego główne zainteresowania naukowe obejmują wykorzystanie informacji wizyjnej w robotyce.

e-mail: tkornuta@ia.pw.edu.pl

Prof. nzw. dr hab. inż. Cezary Zieliński

Jest profesorem nadzwyczajnym Politechniki Warszawskiej na Wydziale Elektroniki i Technik Informacyjnych. W latach: 2002-2005 sprawował na tym wydziale funkcję prodziekana ds. nauki i współpracy międzynarodowej, 2005-2008 zastępcy dyrektora Instytutu Automatyki i Informatyki Stosowanej (IAIS) ds. naukowych, a od 2008 pełni funkcje dyrektora tego instytutu. Od uzyskania habilitacji w roku 1996 pełni rolę kierownika Zespołu Robotyki w IAIS. Od 2007 roku jest członkiem i sekretarzem Komitetu Automatyki i Robotyki Polskiej Akademii Nauk. Od 2008 roku współpracuje z Przemysłowym Instytutem Automatyki i Pomiarów. Jego zainteresowania badawcze koncentrują się na zagadnieniach związanych z programowaniem i sterowaniem robotów.

e-mail: c.zielinski@ia.pw.edu.pl