

ОТКРИВАНЕ НА ОБЕКТИ ОТ ИЗОБРАЖЕНИЕ ЧРЕЗ ИЗПОЛЗВАНЕ НА MATLAB®

Шериф Шериф, Йордан Кралев, Цоньо Славов

Резюме: Откриването на обекти от сцена с голям брой елементи е една от основните задачи в компютърното зрение. Редица изследвания са насочени към оптимизацията на този процес чрез използване на машинно самообучение, където съставянето на алгоритми с конкретни инструкции за решаване на задачата е неприложимо. Повечето вградени системи за откриване на обекти се основават върху алгоритми използващи монохромни изображения. Затова в настоящата работа са изградени модели за преобразуване на цветовете равнини в изображения и са разгледани основните етапи от алгоритъм за откриване на обекти, както и функциите чрез които се осъществява това в средата на MATLAB®.

Ключови думи: модели за преобразуване на цветови равнини, монохромни изображения, откриване на обекти

OBJECTS DETECTION FROM AN IMAGE USING MATLAB®

Sherif Sherif, Jordan Kraleв, Tsonyo Slavov

Abstract: Objects detection from a cluttered scene is one of the main tasks in computer vision. A lot of research has focused on the optimization of this process by using machine learning, where creating algorithms with specific instructions for solving a problem is not applicable. Most of embedded systems for detection object are based on algorithms using monochrome (intensity) images. Therefore, in the article are created models for color space conversion from images and the main stages of the object detection algorithm are discussed, as well as the functions through which this is done in MATLAB

Keywords: models for color space conversion, intensity images, object detection

1. ВЪВЕДЕНИЕ

В днешно време камерите имат основна роля в редица електронни устройства и могат да се видят навсякъде, от мобилните телефони в джоба ни до камерите за наблюдение и микроскопичните камери, използвани в медицината. От няколко години се наблюдава голям възход в компютърното зрение с развитието на раз-

нообразието от техники за изпълнение на определени задачи. Тези задачи включват анализ на движение, реконструкция на сцена, възстановяване на изображение и съвпадение [1, 2, 3, 4]. В този доклад авторите са се съсредоточили върху един конкретен метод, който открива обект от сцена с голям брой елементи, също така е направено и сравнение с други методи, които не са разгледани в настоящата работа с цел да се покажат предимствата и недостатъците. Някои алгоритми работят по-добре с определени набори от данни, докато други не са толкова ефективни при анализирането на едни и същи данни. Алгоритмите за компютърно зрение се използват широко за разпознаване, манипулиране и извличане на детайли от изображение. Тези процеси се извършват с помощта на различни алгоритми и техники. Всеки алгоритъм има свой уникален начин за идентифициране и управление на данните. Критериите за ефективност на алгоритмите се различава за всеки един, въпреки че целта на алгоритъма е една и съща – съвпадение. Съответствието на изображенията е подсистема от компютърното зрение, което се фокусира върху намирането на сходство или множество сходства между набор от изображения и евентуални съвпадения. Тази конкретна задача за съпоставяне на подобни изображения е изпълнена с помощта на алгоритъм, който ще бъде разгледан в следващите параграфи.

2. МОДЕЛИ ЗА ПРЕОБРАЗУВАНЕ НА ЦВЕТОВИТЕ РАВНИНИ В ИЗОБРАЖЕНИЯТА

Първоначално трябва да се дефинира изображение с реални цветове в MATLAB®, като това се реализира по следния начин [5]:

```
Img_RGB = imread('tree, house and cars.jpg');
```

Дефинираното изображение е тримерен масив. Всяка равнина на този масив представлява стойностите на червения, зелен и син цвят на изображението. Изображението, което представя тази матрица може да се види чрез функция *imshow()*. За изграждането на модели за преобразуване се използват наличните функции в Matlab® или блокове от библиотека Computer Vision™ в Simulink®. Ще бъдат разгледани няколко примера, в които входното изображение ще се преобразува в различни цветови равнини.

Преобразуване на изображение с реални цветове в бинарно

За преобразуването в бинарни изображения няма нужда от изграждане на симулационен модел чрез следната вградена функция:

```
Img_Binary = im2bw(Img_RGB);
```

пиксели от входното изображение над определен праг на яркост се заменят със стойност 1 (бяло), а тези под прага със стойност 0 (черно). Прагът на яркост в този пример е стойността сто двадесет и седем, понеже входното изображение е осем битово ($2^8 = 256$).



Фиг. 1. Изображение с реални цветове



Фиг. 2. Бинарно изображение

На Фиг. 1 е показано оригиналното изображение, което съдържа цветовите равнини на трите основни цвята, а на Фиг. 2 е дадено бинарното изображение, което има само една цветова равнина с два цвята – бял и черен.

Преобразуване на изображение с реални цветове в монохромно

Уравнението за преобразуване в монохромни изображения има следния математически запис.

$$I = [0.299 \quad 0.587 \quad 0.114] \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (1)$$

R' , G' и B' са матрици и всяка една от тях представя по една равнина от входящото изображение с реални цветове. Някои алгоритми за обработка на изображения са специфицирани за монохромни изображения.

Преобразуване на изображение с реални цветове в компонентно изображение

Това преобразуване се определя чрез уравнение (2), а уравнение (3) е за преобразуване от компонентно изображение към изображение с реални цветове.

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + A \times \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (2)$$

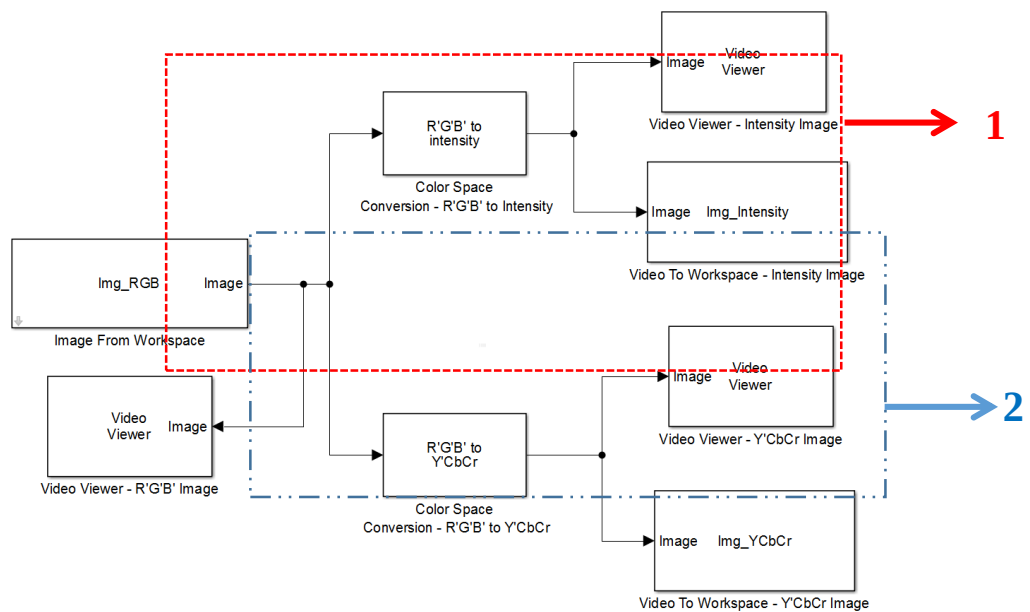
$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = B \times \left(\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right) \quad (3)$$

Стойностите на матрици A и B , показани в Табл. 1, се различават в зависимост от разделителната способност.

Табл. 1 Параметри на матриците

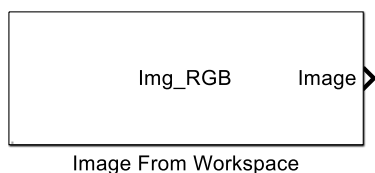
Матрици	Стандартна разделителна способност	Висока разделителна способност
A_L	$\begin{bmatrix} 0.26 & 0.50 & 0.1 \\ -0.15 & -0.29 & 0.44 \\ 0.44 & -0.37 & -0.07 \end{bmatrix}$	-
B_L	$\begin{bmatrix} 1.16 & 0 & 1.61 \\ 1.16 & -0.39 & -0.81 \\ 1.16 & 2.02 & 0 \end{bmatrix}$	-
A_H	-	$\begin{bmatrix} 0.18 & 0.61 & 0.06 \\ -0.10 & -0.03 & 0.44 \\ 0.44 & -0.39 & -0.04 \end{bmatrix}$
B_H	-	$\begin{bmatrix} 0.18 & 0.61 & 0.06 \\ -0.1 & -0.03 & 0.5 \\ 0.44 & -0.4 & -0.04 \end{bmatrix}$

За целта на преобразуванията описани чрез уравнение (1) и (2) е изграден симулационен модел показан на Фиг. 3 от където се вижда, че той се състои от няколко блока, които се използват съответно за вмъкване на изображение от работното пространство (Фиг. 4), блок за преобразуване на цветовете равнини (Фиг. 5), блок за извеждане на изображение/видео сигнал в работното пространство (Фиг. 6) и блок за изобразяване (Video Viewer).

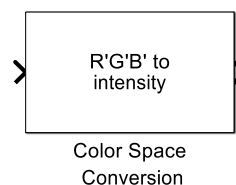


Фиг. 3. Simulink® модел за преобразуване на цветови равнини в изображение

Изграденият модел служи за преобразуване на изображение от цветови равнини на червеният, зелен и син цвят съответно в монохромно изображение (правоъгълник 1) или в компонентно изображение (правоъгълник 2).



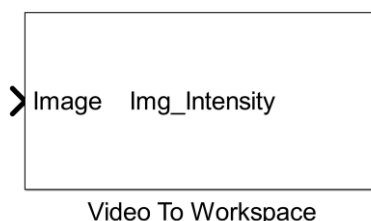
Фиг. 4. Блок за вмъкване на изображение



Фиг. 5. Блок за преобразуване на цветови равнини

Чрез блока показан на Фиг. 4 се вмъква изображение от работното пространство в MATLAB[®]. Ако зададеното изображение е двумерен масив това означава, че на изхода на блока ще имаме бинарно или монохромно изображение. Ако изображението е представено в работното пространство, като тримерен масив, блокът ще изведе цветно изображение, съдържащо информация за трите основни цветни равнини.

На Фиг. 5 е даден блокът за преобразуване на цветовата информация между цветовите пространства, който има два параметъра. Единият позволява да се определи цветовото пространство между които има възможност за преобразуване, а другия параметър е за определяне на типа на въвеждане и извеждане на изображенията, дали да са като един многомерен сигнал или три отделни сигнала за основните цветове.



Фиг. 6. Блок за извеждане на изображение/видео сигнал

Ако изображението (или видео сигнала) е представен със стойности на интензитет (монохромно), то в работното пространство ще се появи тримерен масив, а ако изображението е с реални цветове ще се появи като четиримерен масив. Блокът от Фиг. 6 има четири параметъра: един за задаване на име на променливата; параметър за определяне на броя на входовете в блока (при монохромно изображение се използва един вход, а при реални цветове три входа); останалите два параметъра се използват при извеждане на видео сигнал.

На следващите две фигури е показано оригиналното изображение (Фиг. 1) в две различни цветови равнини. От Фиг. 7 се вижда полученото монохромно изображение, след като е извършен процес по преобразуване на оригинала. На Фиг. 8 е дадено компонентното изображение след преобразуване при стандартна разделителна способност. То съдържа три компонента: на осветеност, разлика в синия и червен цвят.



Фиг. 7. Монохромно изображение



Фиг. 8. Компонентно изображение

3. ОТКРИВАНЕ И РАЗПОЗНАВАНЕ НА ОБЕКТИ ОТ СЦЕНА

Алгоритъмът за откриване на определен обект се базира на намиране на съответствие между точките от основното изображение (сцената/равнината) и еталонното изображение на конкретния обект. Той може да открива конкретните обекти, въпреки промяната в мащаба или завъртането им в равнината, също така е робастен към малки завъртания извън сцената и припокривания на обекти в сцената. Методът работи най-добре за изображения показващи неповтарящи се текстури, което поражда уникални съвпадения на характеристиките. Тази техника вероятно няма да работи добре за изображения съдържащи еднакво оцветени обекти или обекти съдържащи голям брой еднакви елементи.

Преди да се премине към алгоритъма, изображението с реални цветове трябва да се преобразува в монохромно изображение (т. 2) и след това да се запази в един от възможните формати за изображение, което се осъществява чрез функцията *imsave*. След това от него се създава ново изображение съдържащо само еталонния обект. Дефинирането на еталонното изображение съдържащо обекта, който представлява интерес за нас се осъществява така:

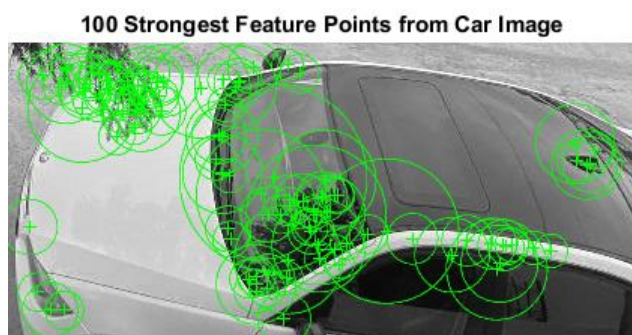
```
Image_Car = imread('Car_Intensity.jpg');
```

От Фиг. 9 се вижда, че избраният обект е автомобил. Еталонното изображение е с ниска резолюция и в работното пространство на MATLAB® представлява масив с размерност 199x418 от осем битов беззнаков целочислен тип. На Фиг. 7 е показано основното изображение, което се декларира по следния начин:

```
Image_Scene = imread('Tree,house and  
cars_Intensity.jpg');
```



Фиг. 9. Еталонно изображение



Фиг. 10. Еталонно изображение с характеристични точки



Фиг. 11. Изображение на сцената с характеристични точки

Вторият етап от алгоритъма е откриването на характеристичните точки в двете изображения [6] – Извадка 1. На Фиг. 10 е показан набора от най-специфични характеристични точки в еталонното изображение, а от Фиг. 11 се вижда изображението на сцената, където отново са показани само най-специфични точки. Процесът по избиране и изобразяване на най-специфичните точки в изображенията се осъществява чрез функция *plot(selectStrongest(points,N))*. В първия аргумент се посочват характеристичните точки на обекта, а във втория се посочва колко на брой от тези най-специфични точки да се изобразят, като въведеното число трябва да е от целочислен тип.

```
Points_Car = detectSURFFeatures(Image_Car);
Points_Scene = detectSURFFeatures(Image_Scene);
```

Извадка 1. Откриване на характеристични точки

След това се извличат характеристиките за описанията, които съдържат информация за характеристичните точки в двете изображения – еталонното и основното изображение (Извадка 2).

```
[Car_Features, Points_Car] = extractFeatures(Image_Car, Points_Car);
[Scene_Features, Points_Scene] = extractFeatures(Image_Scene,
Points_Scene);
```

Извадка 2. Извличане на характеристика за описанията

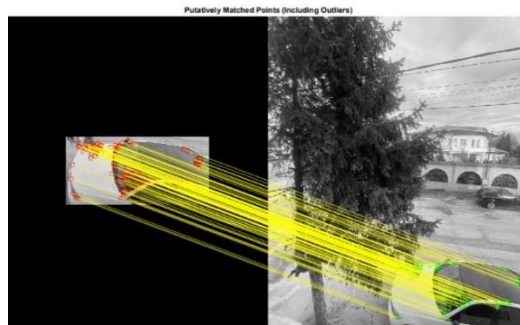
Четвъртата стъпка е съпоставка между характеристиките за описанието на сцената и еталонното изображение.

```
Car_Pairs = matchFeatures(Car_Features, Scene_Features);
```

Изобразяването на предсказаните съвпадащи характеристични точки се реализира чрез функцията

`showMatchedFeatures(Img_1,Img_2, matchedPoints1,matchedPoints2),`

като първите два аргумента са за въвеждане на двете изображения (еталонното и сцената) - представени като масиви в работното пространство [5, 6]. Последните два аргумента са за координатните точки на двете изображения.



Фиг. 12. Съвпадащи точки между изображението на сцената и еталонна

На Фиг. 12 са показани всички съвпадащи характеристични точки между еталонното изображение (в ляво) и основното изображение. Както се вижда няма излишни съвпадащи точки, като такива се класифицират всички точки, които в основното изображение са извън еталона. След това чрез функция *estimateGeometricTransform()* се изчислява трансформацията свързана със съвпадащите точки, като едновременно с това се премахват излишните точки. Освен това чрез тази трансформация се локализира обекта в сцената. Създаването на ограничаващ многоъгълник за еталонното изображение се осъществява по следния начин.

```
Car_Polygon = [1, 1; size(Image_Car, 2), 1; size(Image_Car, 2),  
size(Image_Car, 1); 1, size(Image_Car, 1); 1, 1];
```

Последният етап от алгоритъма е преобразуването на многоъгълника в координатната система на основното изображение. Преобразуваният многоъгълник показва местоположението на обекта в сцената.

```
New_Car_Polygon = transformPointsForward(tform, Car_Polygon);
```

От Фиг. 13 се вижда, че местоположението на еталона е открито успешно в сцената и е поставено в ограничаващия многоъгълник.



Фиг. 13. Местоположение на еталонния обект в сцената

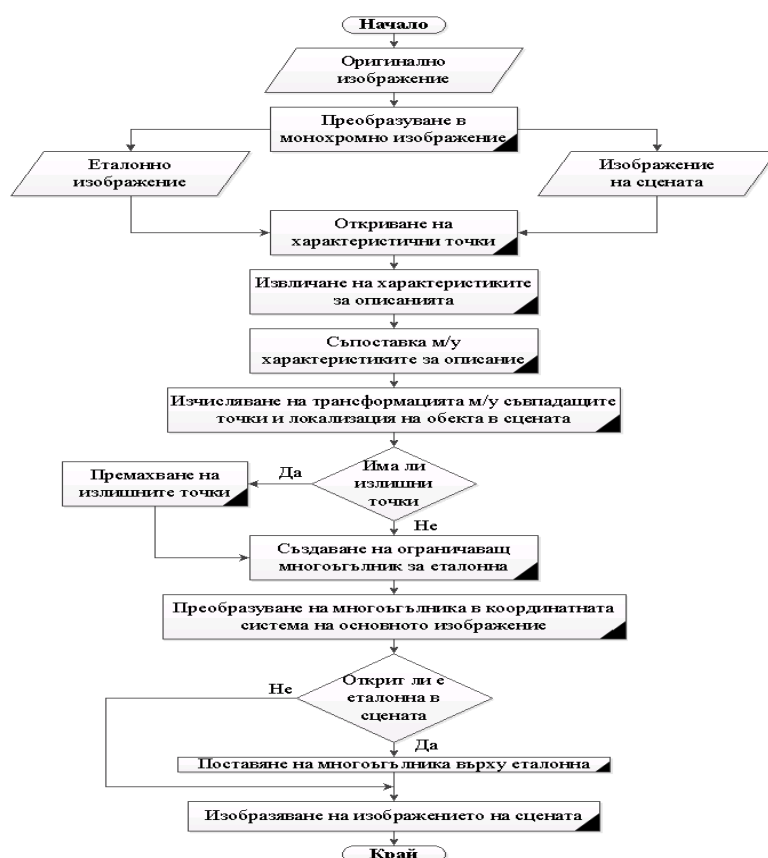
4. ЗАКЛЮЧЕНИЕ

В доклада е разгледан един подход за откриване на конкретен обект от сцена с голям брой елементи. Разгледани са различни модели за преобразуване на цветовете равнини. Подробно са описани всички етапи от алгоритъма за откриване на обект от сцена, както и функциите чрез които се осъществява в MATLAB®, а на Фиг. 14 е дадена и блоковата схема на алгоритъма. Това е доклад, който показва първите стъпки в откриването на обекти, а в бъдещи изследвания ще се направят съпоставки между различни алгоритми и ще се оцени качеството им с цел внедряване на алгоритъма с най-добри показатели в управлението на хуманоиден робот.

Благодарности: Настоящият доклад е реализиран с помощта на финансиране от страна на НИС при ТУ-София по договор 202ПД0013-08.

Табл. 2 Сравнение на алгоритми

Критерии за сравнение	Алгоритъм с характеристични точки	Алгоритъм с разпознаване на шаблони	Алгоритъм с едносвързани цветови региони
Точност	85-95 [%]	40-50 [%]	10-20 [%]
Скорост	Бърз	Среден	Много бавен
Скална инвариация	Да	Не	Не е приложима
Ротационна инвариация	Да	Да	Не е приложима
Сложност	Труден	Среден	Лесен



Фиг. 14. Блокова схема на алгоритъма за откриване на обект от сцена

ЛИТЕРАТУРА

- [1] Bay, H., A. Ess, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008.
- [2] Torr, P., A. Zisserman, MLESAC: A New Robust Estimator with Application to Estimating Image Geometry, CVIU, 2000.
- [3] Sankowski, D., J. Nowakowski, eds. Computer Vision in Robotics and Industrial Applications, World Scientific, 2014.
- [4] Jayanthi, N., S. Indu, Comparison of Image Matching Techniques, International Journal of Latest Trends in Engineering and Technology, Vol. 7, Issue 3, e-ISSN: 2278-621X
- [5] Grimson, W., L. Eric, J. Mundy, Computer vision applications, Communications of the ACM 37.3 (1994), pp. 45-51.
- [6] Kagami, S. High-speed vision systems and projectors for real-time perception of the world, IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, 2010.

Автори: *Шериф Шериф*, докторант, Технически университет-София, Факултет Автоматика, катедра Системи и управление,
e-mail: mr.shark94@abv.bg;

Йордан Кралев, гл. ас. д-р, Технически университет-София, Факултет Автоматика, катедра Системи и управление,
e-mail: jkrlev@tu-sofia.bg;

Цоньо Славов, доц. д-р, Технически университет-София, Факултет Автоматика, катедра Системи и управление,
e-mail: ts-slavov@tu-sofia.bg

Authors: *Sherif Sherif*, Technical University of Sofia, Faculty of Automatics, dept. Systems and Control
e-mail: mr.shark94@abv.bg;

Assist prof. Dr. Jordan Krlev, Technical University of Sofia, Faculty of Automatics, dept. Systems and Control
e-mail: jkrlev@tu-sofia.bg;

Assoc. prof. Dr. Tsonyo Slavov, Technical University of Sofia, Faculty of Automatics, dept. Systems and Control
e-mail: ts-slavov@tu-sofia.bg