

# Rapport Travaux Pratiques 3

**Ecrit par**

**Théodore Lambolez  
Maximilien Petit**

**12 décembre 2016**

# **Table des matières**

# 1 Classification appliquée à la reconnaissance d'objets

**Introduction :** Le but de ce troisième et dernier TP de Traitement Numérique de l'Image est de réaliser une macro sur le logiciel Optimas qui permettra de reconnaître des objets présents sur la scène capturée. Comme toujours, il faut préciser les hypothèses de bases. Ici, nous faisons en sorte que le capteur prenne toujours la scène avec le même angle d'inclinaison et avec la même distance de celle-ci. La macro est présente dans la partie "Annexe".

**Manipulations préalables :** Avant d'utiliser cette macro, il est impératif de s'assurer que la calibration nommée "One\_pixel\_per\_centimeter"

**Méthode mise en oeuvre :** Ici, avant toute chose, nous nous assurons que l'option de calibration nommée "One\_pixel\_per\_centimeter" soit bien sélectionnée.

Ensuite, comme pour les précédentes macros réalisées sur Optimas, nous commencerons par réaliser une binarisation de l'image. Il faut donc définir un seuil, qui devra s'adapter à l'image. C'est pourquoi, on a choisi d'utiliser une nouvelle fois l'outil nommé "auto-threshold". Ici, en raison du nombre d'objets présent sur les images, nous avons choisi d'utiliser l'option de détection du seuil par modélisation d'une exponentielle décroissante.

Une fois l'image binarisée, on constate que certains objets sont séparés en plusieurs ou bien que d'autres présentent des évidements. Afin de faciliter la classification, il est important de mettre en oeuvre une solution pour les rassembler et les reboucher. Nous avons choisi d'utiliser des opérations de morphologie mathématiques pour ce faire. En effet, nous réalisons successivement une fermeture et une ouverture.

Il est maintenant temps de passer à la séparation des objets et à leur identifications. Pour ce faire, on débute en paramétrant le "Data Sampling Area" de telle sorte que la détection de contour se fera en 8-voisinage en prenant en compte les évidements. On spécifiera également une longueur minimale de frontière intérieure de 5 pixels et le fait qu'on ne prend pas en compte les objets présents au niveau des bords de l'image. En effet, si les objets touchent le bord, on ne peut pas être certain que celui-ci est complet. Cela faussera potentiellement son identification.

Ensuite, on utilise l'outil "Particules Count" dans le but de mettre en évidence les différentes régions isolées avec le paramétrage précédant. Ces régions isolées correspondent aux différents objets.

Afin de les identifier, il faut créer une nouvelle classe pour chacun des types d'objets que l'on souhaite identifier et une dernière classe correspondant aux objets que l'on ne

souhaite pas identifier.

Enfin, nous nous assurerons de l'extraction de toutes les données nécessaires à la classification à l'aide du "Measurement Explorer".

Nom classe	Nb Euler	Nb Parents	Circularité	Rectangularité	Région	Largeur
Clés	1		$> 80$		$> 3500$	$< 100$
Rondelles	0	0	$< 30$	$< 0.65$		
Dés face 1	0	0	$< 30$	$< 0.65$		
Dés face 2	-1	0	$< 30$	$< 0.65$		
Dés face 3	-2	0	$< 30$	$< 0.65$		
Dés face 4	-3	0	$< 30$	$< 0.65$		
Dés face 5	-4	0	$< 30$	$< 0.65$		
Dés face 6	-5	0	$< 30$	$< 0.65$		
Objets robotiques	1	0	$18 < x < 22$	$> 0.65$		

TABLE 1.1 – Tableau des critères choisis (hors classe Intrus)

**Critère choisis pour les 9 classes créées :** Pour ce qui est de la dernière classe correspondant aux objets que nous ne souhaitons pas identifier, nous avons simplement pris tous les objets n'appartenant à aucune des classes précédentes.

**Analyse des résultats :** Nous allons dans ce paragraphe analyser les résultats de notre macro pour différentes images.

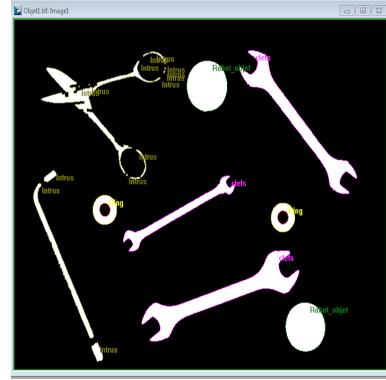
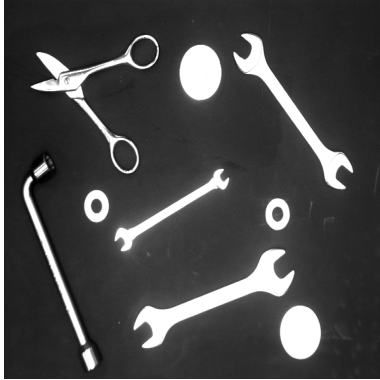


FIGURE 1.1 – Résultat de l'identification d'objets avec la macro sur l'image Objets1.tif

On peut constater sur cette image que tous les objets que l'on veut identifier ont bien été identifiés. Néanmoins, si on voulait identifier les ciseaux, l'étape de reformation des objets ici proposée n'est pas suffisante afin de rassembler toutes les parties de cet objet détachées par la binarisation. On pourrait également tenter d'améliorer le seuil de binarisation choisi.

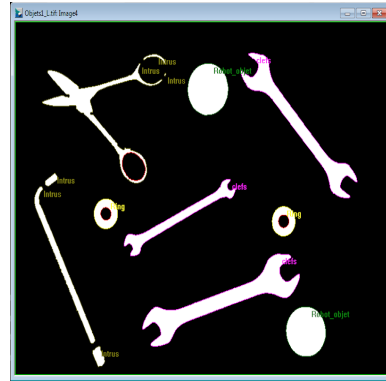
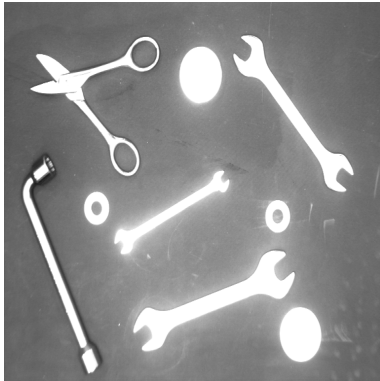


FIGURE 1.2 – Résultat de l'identification d'objets avec la macro sur l'image Objets1\_L.tif

Sur cette image dont la scène est davantage éclairée, on peut constater qu'une bonne partie des ciseaux a été reformée. On constate également que le fort éclairage n'a pas influencé le résultat de notre identification des objets souhaités.

De même, un faible éclaircissement de cette scène n'affecte pas notre résultat.

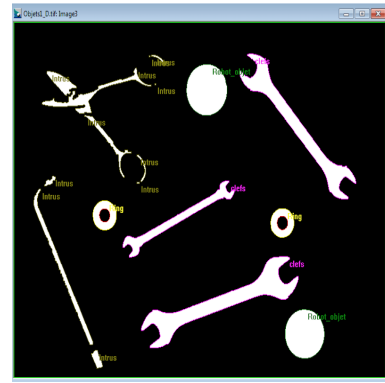
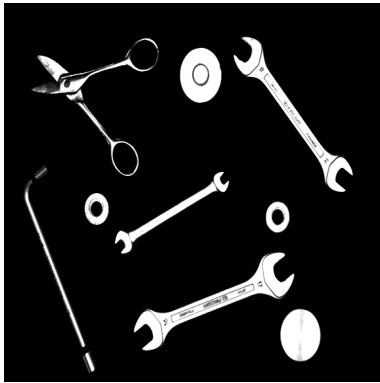


FIGURE 1.3 – Résultat de l'identification d'objets avec la macro sur l'image Objets1-D.tif

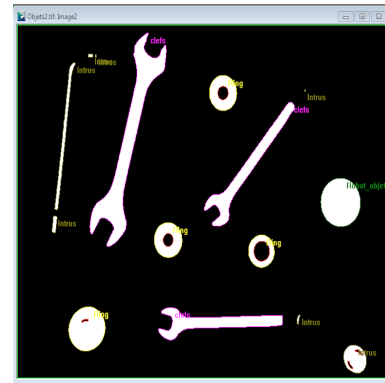
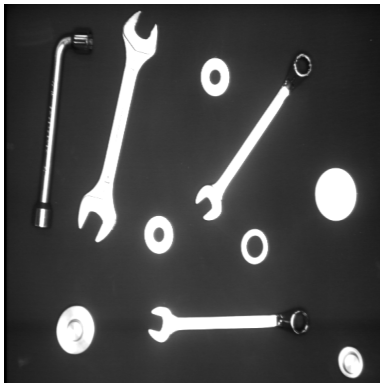


FIGURE 1.4 – Résultat de l'identification d'objets avec la macro sur l'image Objets2.tif

Le résultat de la macro pour cette image apporte une nouvelle limite. En effet, on constate qu'une pièce robotique est identifiée comme une rondelle. Ce problème est dû à un phénomène d'ombre portée. En fait, le fait que le capteur ne soit pas exactement au dessus de cette pièce, fait que le bord est suffisamment important pour être capté. Comme une ombre est plus foncée que l'objet en lui-même, la binarisation globale de l'image revient à considérer ces pixels comme du fond. D'où le nombre d'Euler égal à 0 (1 -1 évidemment) impliquant selon la classification réalisée, qu'on a affaire à une rondelle et non à une pièce robotique.

Ici, le fort éclairage de la scène, augmente le contraste entre les pièces et cette portée. Par conséquent, le phénomène mis en évidence précédemment est amplifié.

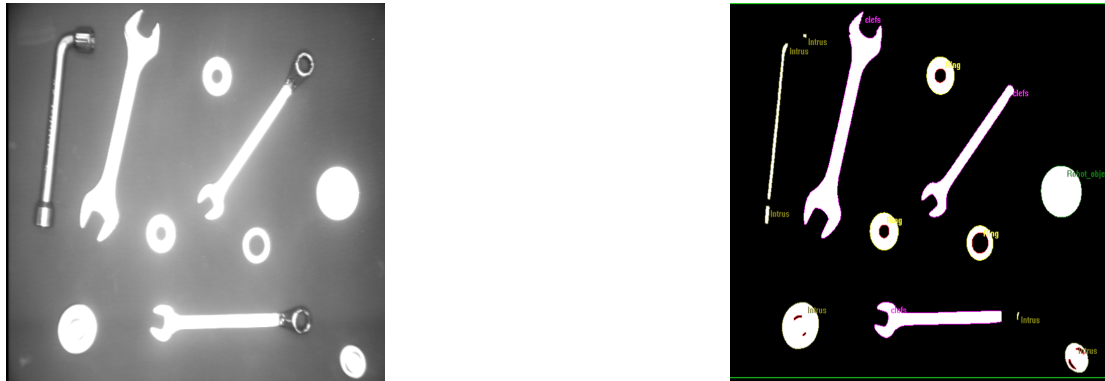


FIGURE 1.5 – Résultat de l'identification d'objets avec la macro sur l'image Objets2\_L.tif

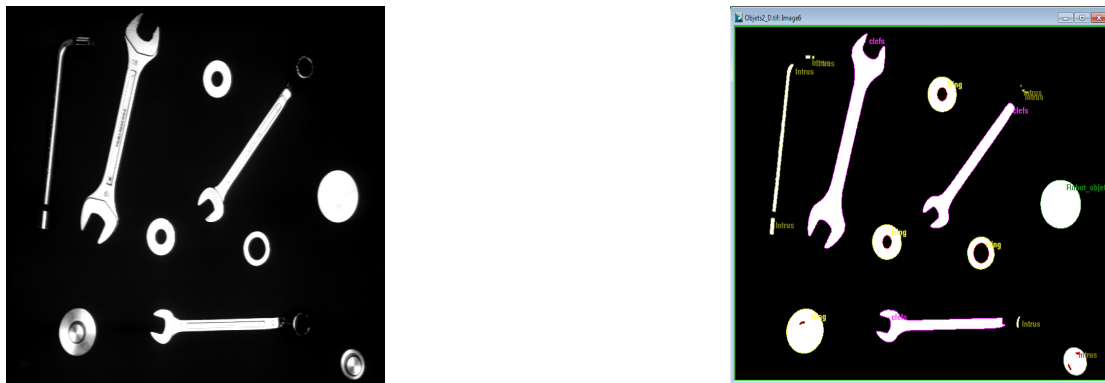


FIGURE 1.6 – Résultat de l'identification d'objets avec la macro sur l'image Objets2\_-D.tif

Dans ce cas où la scène est plus sombre, on constate une légère amélioration. Cependant, elle n'est pas suffisante.

Pour cette image, la macro réalise le travail espéré.

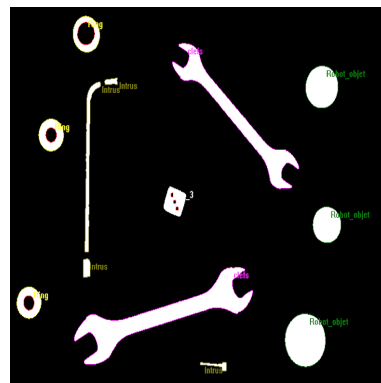
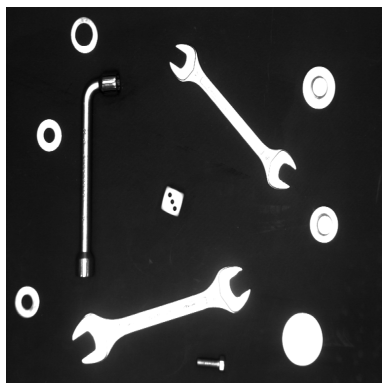


FIGURE 1.7 – Résultat de l'identification d'objets avec la macro sur l'image Objets3.tif

Le résultat sur l'image Objets4.tif, révèle un autre problème dans notre classification. En effet, nous n'avons pas réussi à ne pas identifier la moitié de clé comme n'en étant pas une. Il faudrait réaliser une classification dont les bornes seraient plus rapprochées afin de ne plus l'identifier. Néanmoins, en changeant les valeurs choisies, la classification pour les autres images risque de ne plus être bonne.

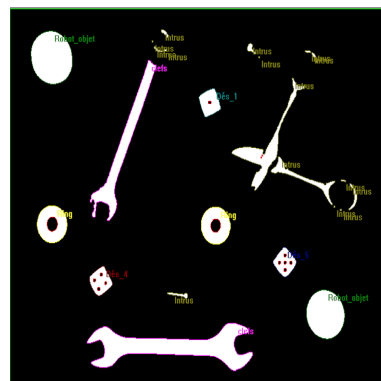


FIGURE 1.8 – Résultat de l'identification d'objets avec la macro sur l'image Objets4.tif



Nous avons le même problème pour le stylet métallique présent sur cette image. Notre classification mène à considérer cet objet comme une clé.

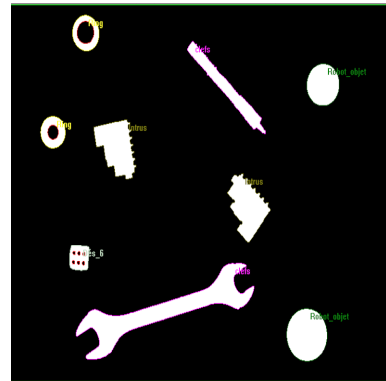
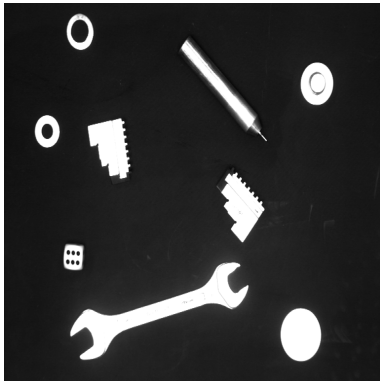


FIGURE 1.9 – Résultat de l'identification d'objets avec la macro sur l'image Objets5.tif

Les derniers résultats présentés n'apportent aucune nouvelle limite à notre macro.

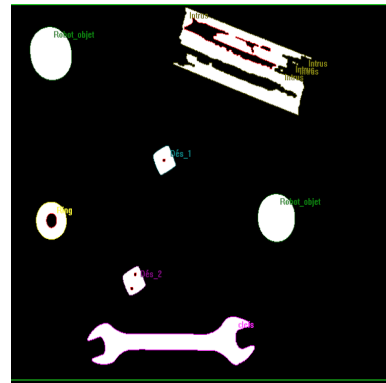
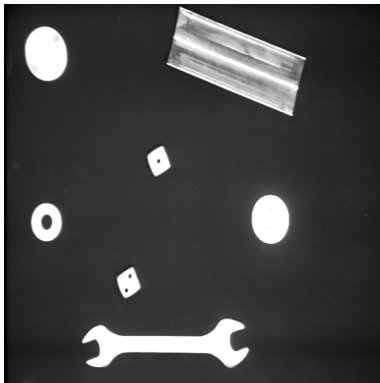


FIGURE 1.10 – Résultat de l'identification d'objets avec la macro sur l'image Objets6.tif

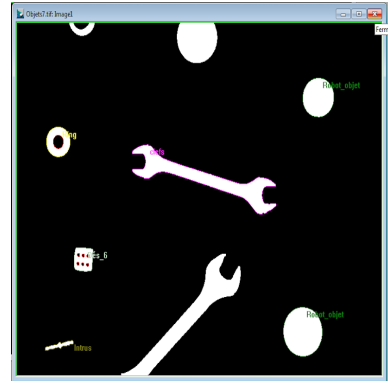
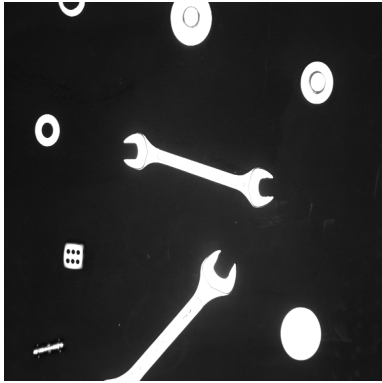


FIGURE 1.11 – Résultat de l'identification d'objets avec la macro sur l'image Objets7.tif

## 2 Annexe

```

ClearScreen ();
Calibrate (One_Pixel_per_Centimeter);
AreaCNVFactors[0..15] = 100.0 : 1.0 : -1.0 : 256.0 : 100.0 : 5.0 : 1.0 : -1.0 : -1.0 : 0.0 : 255.0 : -1.0 :
2.0 : 0.0 : 0.0;
REAL rPairs[1];
Histogram();
//Grab (3);
rPairs = GetAutoThreshold(ArROIHistogram,3,2,10.0,ActiveLuminanceRange);
if ( rPairs && 1 < GetShape(rPairs)[0] )
{
    Threshold(rPairs[1,0]:rPairs[1,1]);
    GrayToBinary();
}
Delete(rPairs);

BINB_iIterations = 1;
// Binary Closing [dilate then erode]
DilateFilter(BINB_iIterations);
ErodeFilter(BINB_iIterations);
Threshold ( 127.5:255.0 );

BINB_iIterations = 1;
// Binary Opening [erode then dilate]
ErodeFilter(BINB_iIterations);
DilateFilter(BINB_iIterations);
Threshold ( 127.5:255.0 );

ActivateMeasurementSet("Area Morphometry Set");
RunMacro ("C:/PROGRAM FILES (X86)/OPTIMAS 6.5/MACSRC/PC/pps.mac");
PPS_CountMacro ();
CloseWindow("Particle Count");

// dÉfinitions des classes
CreateEmptyClass("clefs", 2, 'A'); // classe vide
GetOrSetField("clefs", 302, "clefs"); // le label
GetOrSetField("clefs", 306, "Ar_clefs_Member_A = ArCircularity > 80 && ArArea > 3500 && ArBreadth < 100 ;"); //
la dÉfinition

CreateEmptyClass("Ring", 2, 'B');
GetOrSetField("Ring", 302, "Ring");
GetOrSetField("Ring", 306, "Ar_Ring_Member_B = ArEulerNumber == 0 && ArNestNoParents == 0 && ArRectangularity >
0.65;");

CreateEmptyClass("Des_1", 2, 'C');
GetOrSetField("Des_1", 302, "Des_1");
GetOrSetField("Des_1", 306, "Ar_Des_1_Member_C = ArEulerNumber == 0 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Des_2", 2, 'D');
GetOrSetField("Des_2", 302, "Des_2");
GetOrSetField("Des_2", 306, "Ar_Des_2_Member_D = ArEulerNumber == -1 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Des_3", 2, 'E');
GetOrSetField("Des_3", 302, "Des_3");
GetOrSetField("Des_3", 306, "Ar_Des_3_Member_E = ArEulerNumber == -2 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Des_4", 2, 'F');
GetOrSetField("Des_4", 302, "Des_4");
GetOrSetField("Des_4", 306, "Ar_Des_4_Member_F = ArEulerNumber == -3 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Des_5", 2, 'G');
GetOrSetField("Des_5", 302, "Des_5");
GetOrSetField("Des_5", 306, "Ar_Des_5_Member_G = ArEulerNumber == -4 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Des_6", 2, 'H');
GetOrSetField("Des_6", 302, "Des_6");
GetOrSetField("Des_6", 306, "Ar_Des_6_Member_H = ArEulerNumber == -5 && ArNestNoParents == 0 &&
ArRectangularity < 0.65 && ArCircularity < 30;");

CreateEmptyClass("Robotic_object", 2, 'I');
GetOrSetField("Robotic_object", 302, "Robot_objet");
GetOrSetField("Robotic_object", 306, "Ar_Robotic_object_Member_I = ArNestNoParents == 0 && ArEulerNumber == 1
&& ArCircularity < 22 && ArCircularity > 18 && ArRectangularity > 0.65 ;");

CreateEmptyClass("Intrus", 2, 'J');
GetOrSetField("Intrus", 302, "Intrus");
GetOrSetField("Intrus", 306, "Ar_Intrus_Member_J = !Ar_clefs_Member_A && !Ar_Des_1_Member_C && !
Ar_Des_2_Member_D && !Ar_Des_3_Member_E && !Ar_Des_4_Member_F && !Ar_Des_5_Member_G && !Ar_Des_6_Member_H && !
Ar_Ring_Member_B && !Ar_Robotic_object_Member_I && ArNestNoParents == 0;");

// extraction des caractÉristiques et des Étiquettes
SetExport(mArCircularity,1,TRUE);
SetExport(mArEulerNumber,1,TRUE);
SetExport (mArNestNoParents, optExtractFlag, TRUE);
SetExport (ArNestNoParents, optExtractFlag, TRUE);
SetExport (mArRectangularity, optExtractFlag, TRUE);
SetExport (ArRectangularity, optExtractFlag, TRUE);
SetExport (mArBreadth, optExtractFlag, TRUE);
SetExport (ArBreadth, optExtractFlag, TRUE);
MultipleExtract(TRUE);

```

FIGURE 2.1 – Macro de reconnaissance de forme