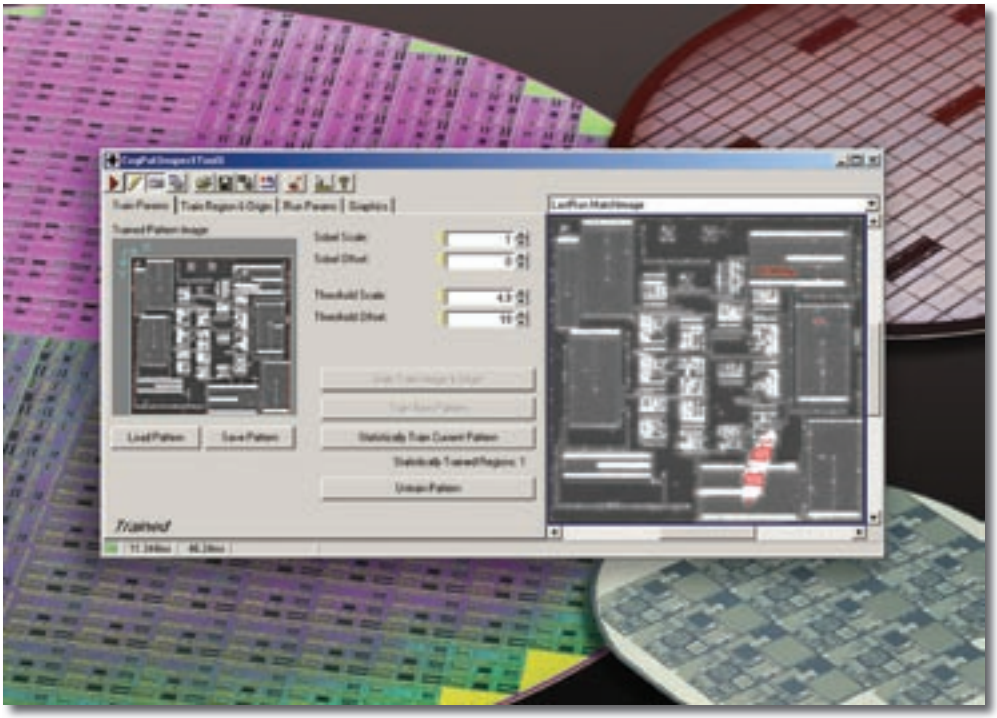
Les outils logiciels de vision industrielle

En vision industrielle, la question des performances des outils logiciels et de leur adaptation au niveau d'expertise des utilisateurs, est au moins aussi importante que celle de l'éclairage ou de la qualité de l'image. Cet article propose un petit tour d'horizon des différents outils que doivent intégrer les solutions logicielles de vision. Il revient également sur les différentes approches existantes en matière de programmation.

Depuis une dizaine d'années, la Vision Industrielle s'est largement démocratisée, comme en témoigne l'évolution globale du marché, aussi bien en terme de nombre d'unités vendues, qu'en terme de prix moyen des systèmes. Etant donnée la grande variété d'applications envisageables, allant du guidage robotique à la reconnaissance de caractères, en passant par le contrôle qualité, les systèmes de vision doivent offrir la plus grande flexibilité possible. Comme dans beaucoup d'autres disciplines, la partie logicielle revêt à ce titre une importance capitale. Les concepteurs, qu'il s'agisse d'utilisateurs finaux non-spécialistes de la vision, ou bien encore d'intégrateurs confirmés, doivent en effet disposer d'outils logiciels adaptés leur permettant de réaliser aussi bien des études de faisabilité rapides, que des développements plus poussés, sans avoir en permanence à réinventer la roue. Pour cela, les logiciels doivent non seulement inclure de nombreux outils et algorithmes, nécessaires à la réalisation de diverses tâches de vision, mais également offrir différents niveaux d'utilisation, adaptés au degré d'expertise de chaque profil d'utilisateur. De nombreuses solutions commerciales répondant plus ou moins bien à l'ensemble de ces besoins sont aujourd'hui disponibles sur le marché. On distingue principalement deux types d'approches : paramétrables ou programmables, sur lesquelles nous reviendrons par la suite. Mais concentrons-nous dans un premier temps sur les principales fonctionnalités que doit intégrer tout logiciel de vision qui se respecte.

Les pré-traitements

En vision industrielle, aucun traitement ne saurait rattraper une mauvaise qualité d'image. Cependant, si bonne soit-elle, une image brute est rarement exploitable telle quelle. L'analyse des informations qu'elle contient nécessite souvent que celle-ci soit modifiée, soit pour en améliorer la qualité, en éliminant le bruit par exemple, soit pour préparer l'image en vue de traitements ultérieurs : en faisant ressortir certains éléments caractéristiques, en séparant les objets d'intérêt du fond, en éliminant les éléments en contact avec la bordure de l'image… Les outils de prétraitement traditionnellement disponibles en standard dans une solution logicielle de vision industrielle incluent des LUTs (Look-Up Table) et des filtres spatiaux et/ou fréquentiels.



Une LUT est l'opérateur le plus simple que l'on puisse trouver en traitement d'images. Elle permet de convertir les niveaux de gris d'une image en appliquant une fonction à chacun des pixels. Son utilisation permet notamment d'améliorer le contraste et la luminosité des zones sombres d'une image. Ainsi, pour éclaircir une image, on applique la fonction logarithme à chaque niveau de gris. A l'inverse, pour rendre plus sombre une image un peu trop saturée, on applique une fonction exponentielle. On peut remarquer que le seuillage n'est rien d'autre qu'une transformation LUT particulière, celle qui associe le noir à tous les niveaux de gris inférieurs à un certain seuil, et le blanc à tous les autres.

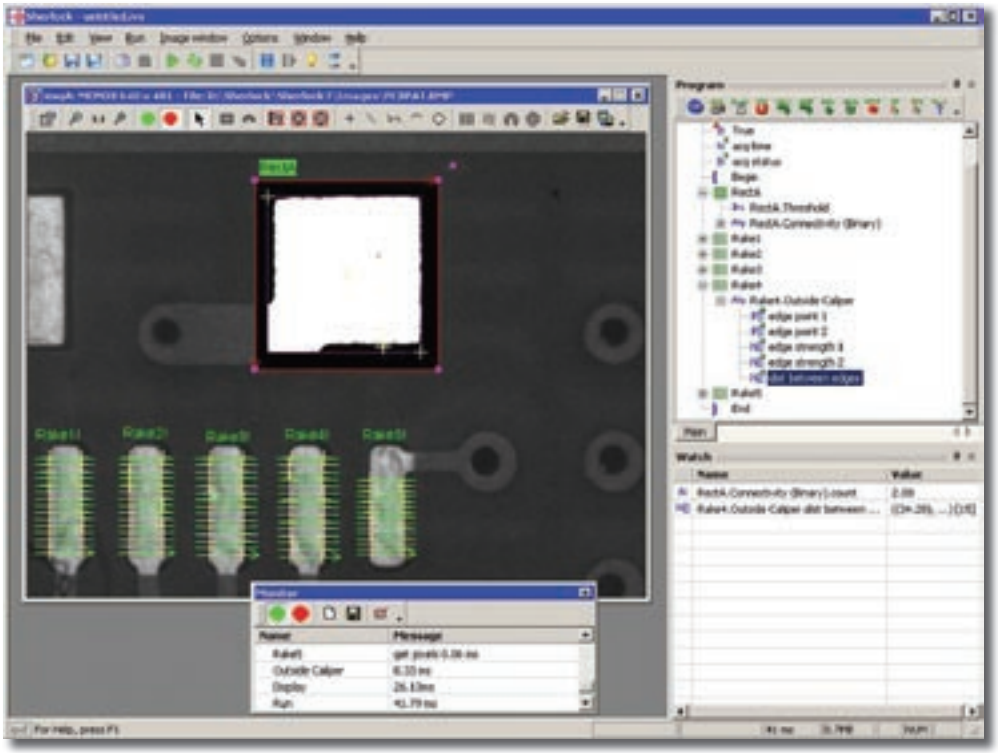
Plus complexes, les opérateurs de filtrage sont nombreux et permettent de réaliser quantité de traitements sur une image brute. Ils sont principalement utilisés pour révéler les contours des objets, éliminer le bruit, adoucir ou accentuer les détails. Ces opérateurs agissent soit dans le domaine spatial, en recalculant la valeur de chaque pixels en fonction des valeurs des pixels avoisinants, soit dans le domaine fréquentiel, en agissant non pas sur l'image brute, mais sur sa représentation fréquentielle. Celle-ci est généralement obtenue par le biais d'une Transformation de Fourier Rapide (FFT pour Fast Fourier Tranfsorm).

Outils de localisation

La localisation d'un objet ou d'une forme au sein d'une image est souvent l'une des premières opérations réalisées dans une application de vision industrielle. Les principales approches utilisées par les logiciels de vision pour la localisation d'objets sont basées sur la détection de contours ou la reconnaissance de formes. Typiquement, dans la première approche, le logiciel analyse les pixels situés le long d'un réseau de profils, qui peut être constitué de lignes droites parallèles ou concourantes (spoke), de portions de cercles concentriques, etc. Il détecte ainsi les transitions présentes dans l'image le long de ces profils, ce qui permet par exemple de localiser les contours d'un objet. La seconde approche, plus complexe, est basée sur la détection dans l'image d'ensembles de pixels coïncidant avec un modèle connu *a priori* ou « appris » par le système. La reconnaissance de formes peut porter aussi bien sur des caractéristiques de texture, que sur des caractéristiques géométriques de l'objet recherché. Traditionnellement, les outils de reconnaissance de formes étaient basés sur la technique désormais bien connue d'Inter Corrélation Normalisée (NCC pour Normalized Cross Correlation). Simple à implémenter et à accélérer, celle-ci souffre cependant d'importantes limitations. Tout d'abord, elle impose que la cible apparaisse toujours dans le champ de vision de la camera avec approximativement la même orientation, et à la même distance. Pour la même raison, la NCC est incapable de localiser des objets plus grands ou plus petits que le modèle de référence, même si la forme géométrique est identique. Par ailleurs, la NCC ne peut pas travailler sur des modèles non-rectangulaires. Cela signifie que, quelle que soit la forme de l'objet à retrouver dans l'image, l'apprentissage ne pourra être effectué, au mieux, que sur le plus petit rectangle contenant l'objet ; ce qui rend impossible la détection d'objets qui se touchent, ou encore la gestion de fonds variables. Enfin, la technique NCC échoue lorsque la luminosité varie. Des techniques plus récentes basées sur la géométrie permettent de surmonter ces difficultés. Insensibles aux rotations et au facteur d'échelle, ces techniques peuvent s'adapter à de nombreuses contraintes, telles que les changements non-linéaires d'éclairage, l'inversion vidéo, le bruit, l'occlusion, les objets en contact.

Outils de mesure

De nombreuses applications de vision nécessitent que des mesures précises soient effectuées dans l'image, par exemple pour déterminer si un produit a été correctement fabriqué ou non, s'il rentre ou non dans les limites de tolérances imposées.



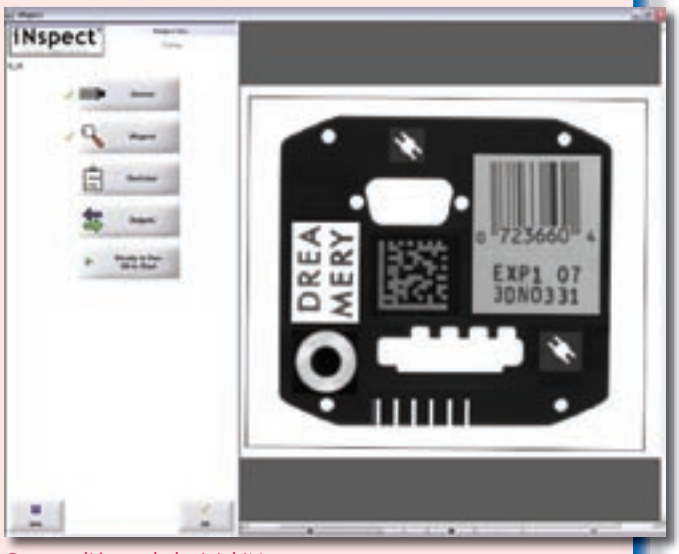
Capture d'écran du logiciel Sherlock 7 d'IPD.

La vision industrielle représente un intérêt particulier notamment dans le cas de petites pièces fabriquées en grandes séries, pour lesquelles la réalisation d'une inspection visuelle par un opérateur s'avère fastidieuse, consommatrice de temps et finalement peu efficace. Les mesures effectuées dans l'image peuvent être de natures diverses. Typiquement, parmi les possibilités

Une application d'inspection en moins de 15 minutes

Le logiciel iNspect, développé par IPD, filiale du groupe Dalsa Digital Imaging, est le parfait exemple d'outil paramétrable s'adressant à des automaticiens non spécialistes de la vision. Commercialisé en France par la société Imasys, cet utilitaire livré en standard avec les automates de vision de la série VAXX, permet de configurer et déployer une application d'inspection en seulement quelques minutes, au travers d'une approche séquentielle jalonnée de cinq grandes étapes. La programmation peut être effectuée soit localement, soit à distance au moyen d'une connexion Ethernet et d'un navigateur web. iNspect offre la possibilité de connecter jusqu'à deux caméras monochromes qui seront déclenchées de façon synchrone, par le biais d'un trigger interne ou externe. Le logiciel donne la possibilité de régler différents paramètres de déclenchement (intervalles, durées, offset), pour piloter les prises de vue mais également l'éclairage, et d'ajuster les paramètres d'exposition, de luminosité et de contraste. iNspect permet d'exprimer les grandeurs mesurées dans l'image en unités du monde réel, par le biais d'un calibrage. Celui-ci peut-être réalisé à l'aide d'un damier, qu'il suffit de présenter à plat, perpendiculairement à la caméra, en spécifiant dans l'unité choisie la valeur de la hauteur des carrés.

Différentes options de recalage sont proposées, en translation et rotation, déterminées à partir de transitions ou d'un modèle (Texture ou Forme). Les performances (temps et variation) du recalage varient en fonction du type de recalage choisi. Pour le contrôle proprement dit, iNspect intègre dix-sept outils : mesures, lecture de caractères, pattern matching, lecture code barre, data matrix, analyse/comptage de forme (Blob), moyenne, écart type, etc. Une fois que les outils nécessaires à l'application ont été positionnés et initialisés, l'utilisateur fixe les règles de décision, décide de différentes actions à mener en fonction des résultats retournés durant l'inspection. Il ne reste plus qu'à affecter les résultats aux différentes sorties TOR, Ethernet, Liaison automate. L'application d'inspection est alors prête à être mise en service.



Capture d'écran du logiciel iNspect.

offertes par les outils disponibles sur le marché, on citera : la mesure de la distance entre deux points de l'image, la détermination de l'angle entre deux lignes représentées par trois ou quatre points, le calcul de la surface d'une forme géométrique, etc. Dans beaucoup d'applications, les mesures effectuées peuvent être directement comparées avec une valeur standard ou avec le résultat d'une mesure identique effectuée dans une image de référence. Toutefois, certaines applications peuvent nécessiter que les résultats de mesures soient comparés avec des valeurs spécifiées en unités réelles. Pour cela, les valeurs mesurées dans l'image doivent pouvoir être converties en utilisant des outils de calibrage, capables de compenser les effets de perspective dus à une orientation non optimale de la caméra, ou encore les distorsions géométriques générées par l'objectif.

Lecture de codes et Reconnaissance de caractères

La plupart des logiciels de vision intègrent désormais des capacités de lecture de codes 1D et 2D, de type code à barres ou DataMatrix. Le processus de lecture se décompose généralement en deux étapes : une phase d'apprentissage durant laquelle l'utilisateur définit la zone de l'image dans laquelle se trouve le code à déchiffrer, suivie d'une phase de reconnaissance durant laquelle la région spécifiée est analysée. De la qualité de l'image dépend évidemment le succès de la reconnaissance, plusieurs facteurs pouvant conduire à l'échec : image bruitée, résolution trop faible, contraste le long des barres, etc. Notons que certains outils permettent de vérifier la qualité du code par rapport à des normes établies, fournissant des informations précieuses sur les performances du processus de marquage. Un grand nombre de logiciels de vision proposent également des solutions de reconnaissance de caractères (OCR pour Optical Character Recognition), par exemple pour la lecture de codes alphanumériques sur des pièces automobiles ou encore pour la vérification des dates et des codes de lot sur des flacons de médicaments. Là aussi, le processus se décompose en deux étapes : apprentissage et reconnaissance. Durant la phase d'apprentissage, l'utilisateur présente au logiciel des modèles de caractères qui lui serviront de base de comparaison durant la phase de reconnaissance. Notons que certains logiciels proposent des fonctionnalités évoluées comme la configuration statistique des polices, qui permet de créer un modèle unique ou une «image de référence» à partir d'une série d'images. Le logiciel peut ainsi mieux traiter les variations normales qu'il rencontrera éventuellement dans la qualité d'impression, que ce soit un contraste faible, une position différente, des dégradés, ou une largeur de trait variable. A moins d'être sûr que le niveau de qualité des étiquettes soit parfaitement constant et conforme au modèle, la possibilité de créer un modèle statistique peut jouer un rôle crucial dans la réussite d'une application de reconnaissance de caractères.

Programmable vs paramétrable

Pour schématiser, il existe deux grandes catégories de logiciels de vision : d'un côté, les logiciels qui fournissent un accès à des algorithmes et fonctions de bas niveau au travers d'environnements de développement standards, basés sur des langages tels que le C, le C++, Java, Visual Basic, LabVIEW, etc. et de l'autre, les logiciels basés sur une approche « paramétrable », qui permettent à l'utilisateur de concevoir et déployer une application de vision au travers d'une interface conviviale, sans avoir à écrire la moindre ligne de code. Les principaux avantages que présentent les outils basés sur la première approche sont évidemment la flexibilité et la portabilité. En effet, lorsque l'on s'occupe d'écrire la totalité d'un programme dans un langage tel que le C++, rien n'est jamais figé : tout peut être personnalisé, adapté, amélioré, modifié… Par ailleurs, une fois écrit, le code peut être utilisé par n'importe quelle application compatible. Toutefois, ce type d'outils présente l'inconvénient de nécessiter une connaissance approfondie du langage utilisé, ainsi que de fortes compétences en développement d'applications de vision. Ils s'adressent donc plutôt à des utilisateurs expérimentés, spécialistes de la vision et maîtrisant parfaitement un ou plusieurs langages de programmation. A l'inverse, les outils basés sur l'approche « paramétrable » sont conçus pour permettre à des développeurs n'ayant pas nécessairement de grande expérience du codage, de générer rapidement un programme de vision. Un grand nombre d'outils de ce type fait appel pour cela à une programmation de type « graphique », basée sur des mécanismes simples et naturels tels que : sélection d'icônes à l'aide de la souris, manipulations de blocs, glisser-déposer, etc. Outre la rapidité avec laquelle ils permettent de créer une application de vision, ces outils présentent l'avantage de donner lieu à des programmes dont la structure est beaucoup plus lisible. Le principal inconvénient de ces solutions est leur manque de flexibilité, notamment lorsqu'un haut niveau de customisation est requis ou lorsque l'application nécessite le développement d'outils de vision spécifiques. Chacune de ces deux approches a ses mérites. Quoi qu'il en soit, étant donnée la grande variété des applications de vision, un logiciel qui ne repose que sur l'une ou l'autre peut vite s'avérer insuffisant, et il n'est pas rare que les développeurs finissent par utiliser plusieurs outils différents, pour combler les manques de chacun. Dans ce cas, l'idéal est de disposer d'un logiciel de vision proposant différents niveaux de développement au sein d'une seule et même plate-forme intégrée. L'environnement de programmation peut alors être sélectionné en fonction des contraintes imposées par l'application. En cas de besoin, les programmes peuvent être migrés d'un environnement de développement à un autre de façon totalement transparente et sans aucune perte.

Evaluez, implémentez, déployez

Cognex Corporation vient d'annoncer des améliorations apportées à VisionPro, son environnement de développement d'applications de vision sur PC. Il s'agit d'un ensemble d'outils interactifs permettant d'évaluer, d'implémenter et de déployer rapidement des applications multi-caméras sans avoir recours à la programmation. Ainsi, l'environnement QuickBuild permet de définir la chaîne d'acquisition d'image, les outils de vision et les critères d'acceptation ou de rejet.

L'outil Communication Explorer envoie les résultats d'inspection sur les lignes d'entrées/sorties, en direction des équipements de communication, automates et autres mécanismes de rejet. Une option de Scripting permet d'étendre une application créée avec QuickBuild sans avoir à la réécrire complètement. L'Application Wizard génère, quant à elle, automatiquement une application runtime avec interface opérateur en l'espace de quelques minutes.

CONDITION	CAUSE PROBABLE	TEST
Rotation de la pièce	✓ Mauvaise fixation de la pièce ✓ Mouvements de la ligne/vibrations	Faire pivoter la pièce entre 0 et 360°
Variation d'échelle	✓ Variation de la distance Caméra/Pièce	Rapprocher et éloigner la caméra de la pièce
Image floue	✓ Optique mal ajustée	Régler la caméra de sorte que la pièce soit floue
Faible contraste	✓ La pièce se fond dans le décor ✓ La pièce a des contours faiblement marqués	Présenter la pièce sur un fond de même couleur
Eclairage changeant	✓ Eclairage variable ou inadapté ✓ Eclairage naturel du soleil	Augmenter et diminuer l'éclairage, ouvrir et fermer le diaphragme
Ombres	✓ Bras du robot ou autre équipement situé au-dessus de la pièce	En utilisant votre main ou un autre objet, créer une zone d'ombre sur la pièce
Pièces superposées	✓ Pièces multiples sans fixation circulant sur la ligne	Recouvrir une partie de la pièce avec un autre objet
Réflexions	✓ Lumière parasite se réfléchissant sur la pièce	Projeter une lumière vive à différents endroits de la surface de la pièce
Vibrations	✓ Mouvements du convoyeur ✓ Présence de moteurs et autres équipements	Sous la caméra, secouer légèrement la pièce
Variations du process	✓ Variabilité du process	Présenter plusieurs pièces d'aspects différents

Comment estimer la précision et la fiabilité des outils de localisation ? (Source Cognex).