

« Dynamic Vision App for HALCON »

Du traitement d'images HALCON à l'application de vision IDS NXT en quelques étapes



Exécutez votre traitement d'images développé avec HALCON directement dans une caméra IDS basée sur une application de vision sans avoir à programmer une application de vision IDS NXT supplémentaire à cet effet. Développer des applications de vision en C++, une activité coûteuse en temps, n'est plus nécessaire ! Et ce, grâce à la « Dynamic Vision App for HALCON ». Elle est le complément parfait pour vos travaux de développement en matière de traitement d'images.

Avec ce conseil technique, vous apprendrez comment créer une bibliothèque HALCON appropriée dans HDevelop en quelques étapes et comment la télécharger et l'exécuter simplement dans un IDS NXT vegas via l'IDS NXT Cockpit, sans écrire une seule ligne de code C++. Dans l'exemple de traitement d'images « *example_code2d.hdpi* », des codes matriciels sont recherchés dans les données d'image capturées et marqués dans les images issues de l'application de vision. Le contenu du code est également sorti dans un format texte. Le framework IDS NXT développé par IDS vous permet de développer sur mesure des applications de vision en choisissant librement les opérations à exécuter sur les caméras basées sur une application de vision. Pour le développement et l'exécution de votre application de vision individuelle, vous êtes pris en charge par l'IDS NXT Vision App Development Kit, qui exige uniquement des connaissances en C++. Mais les choses sont beaucoup plus simples avec la « Dynamic Vision App for HALCON » qui vient d'être développée. Elle est le complément parfait pour vos travaux de développement en matière de traitement d'images. Si vous voulez uniquement exécuter le traitement d'images avec l'HALCON déjà pré-installé, cette application générique prend entièrement en charge la création, coûteuse en temps, des applications de vision en C++. Elle exécute non seulement le traitement d'images HALCON écrit dans HDevelop, mais fournit également les paramètres d'entrée et de sortie définis via l'interface REST. La « Dynamic Vision App for HALCON » affiche aussi ces

paramètres dans l'IDS NXT Cockpit, vous permettant de configurer et de contrôler commodément votre traitement d'images.

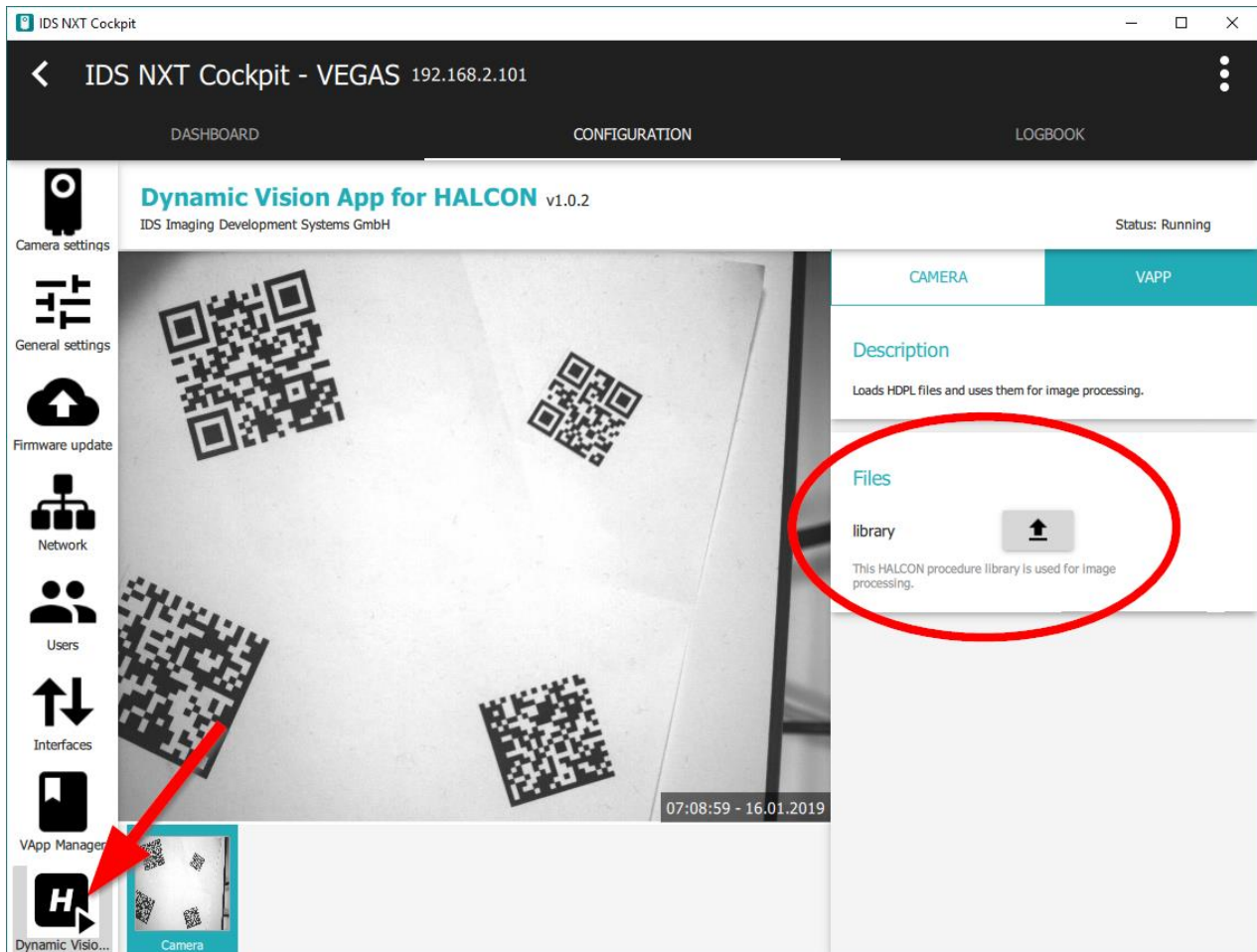


Figure 1 Exécution directe de bibliothèques HALCON.

Conditions préalables

Pour préparer la procédure décrite ici et pour mieux comprendre le concept IDS NXT, il est recommandé de consulter le [manuel « IDS NXT - Développement des applications de vision »](#) et les [didacticiels vidéo IDS NXT](#). Pour la procédure décrite dans le conseil technique, vous avez besoin des éléments suivants :

- IDS NXT vegas (firmware à partir de la version 1.3)
- Windows PC (avec une connexion réseau à IDS NXT vegas)
 - Licence pour l'environnement de développement HALCON 13
 - IDS NXT Cockpit (firmware à partir de la version 1.2.4)
- « Dynamic Vision App for HALCON »

Installez et activez la « Dynamic Vision App for HALCON » sur l'IDS NXT vegas via l'IDS NXT Cockpit. L'installation d'une application de vision est décrite dans le chapitre « Gestionnaire des applications de vision » dans le [manuel « IDS NXT - Fonctionnement »](#).

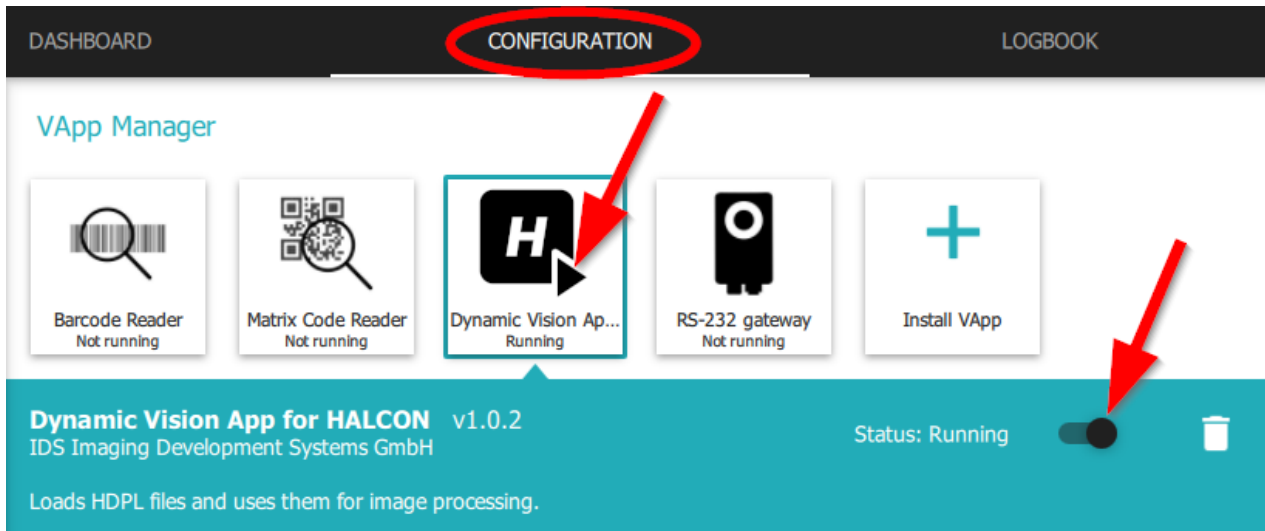


Figure 2 Démarrage de la « Dynamic Vision App for HALCON » dans le gestionnaire des applications de vision, VApp Manager

Si le démarrage de l'application de vision s'est déroulé correctement, la boîte de dialogue de configuration apparaît dans le panneau de configuration de l'IDS NXT Cockpit. Vous pouvez également télécharger des bibliothèques HALCON compatibles, dont le développement est décrit ci-après.



L'installation et la désinstallation des applications de vision IDS NXT ne peuvent être effectuées que par l'utilisateur administrateur !

Développement HALCON

La « Dynamic Vision App for HALCON » établit la connexion entre une « bibliothèque de procédure HDevelop » (HDPL) et le framework IDS NXT dans une caméra basée sur une application de vision de la gamme IDS NXT. Les paramètres de commande et les résultats du traitement d'images sont alors mis à disposition de l'interface REST de la caméra et de l'IDS NXT Cockpit. Le traitement d'images peut ainsi être contrôlé directement comme une application de vision complète.

Assurer la compatibilité de la bibliothèque HALCON avec l'application de vision nécessite l'existence de procédures HALCON avec des noms spéciaux, dont les paramètres d'entrée et de sortie (interface de procédure) sont définis exactement (types de sémantique). C'est alors seulement que la « Dynamic Vision App for HALCON » peut sélectionner automatiquement les variables de configuration ou de résultat appropriées qui seront fournies via l'interface REST.

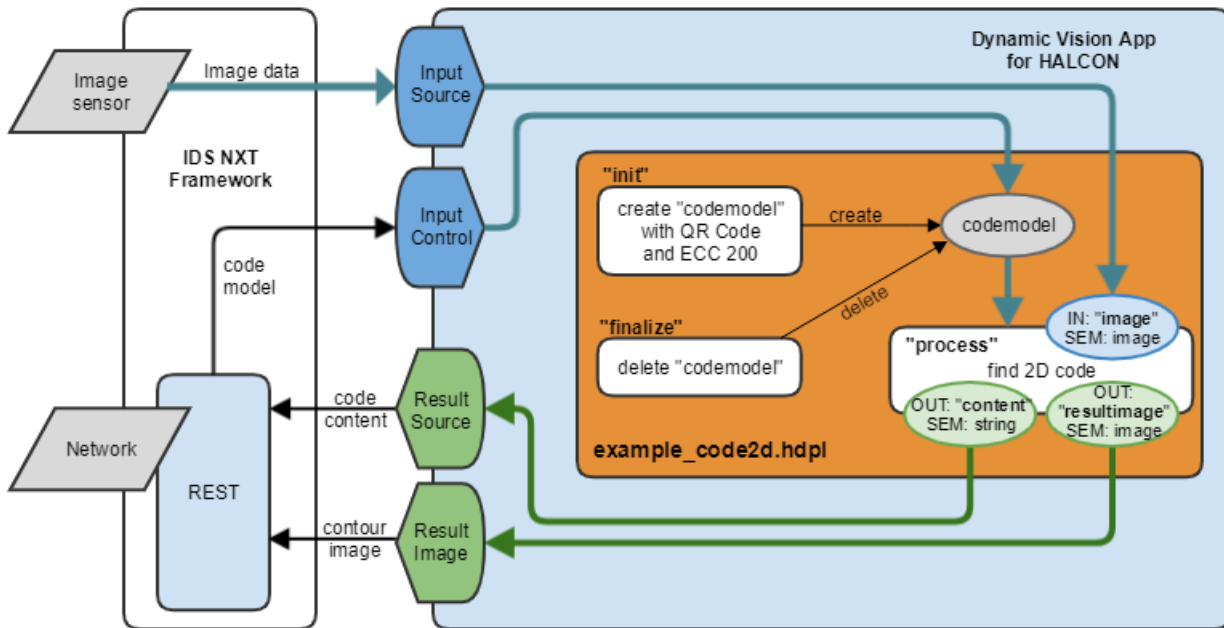


Figure 3 La « Dynamic Vision App for HALCON » établit la connexion entre HALCON et l'IDS NXT Framework.

Structure de la bibliothèque HALCON

La communication entre l'application de vision et le traitement d'images HALCON exige les procédures HDPL suivantes comme points d'entrée :

- « **init** » - est exécuté au chargement de la bibliothèque. Les paramètres de contrôle des entrées, **InputCtrlParams** (variables, paramètres de commande), et les paramètres de contrôle des sorties, **OutputCtrlParams** (résultats), sont enregistrés ici.
- « **finalize** » - est appelé avant le déchargement de la bibliothèque et nettoie toutes les structures de données créées par **init**.
- « **train** » - active la fonction d'apprentissage du traitement d'images HALCON. Cette procédure est facultative. Lorsqu'elle est disponible, l'apprentissage correspondant est exécuté.
- « **process** » - est la fonction de traitement du traitement d'images proprement dit.

Création de procédures

Créez les procédures de type « procédure de bibliothèque(.hdpl) » requises dans HDevelop et enregistrez-les dans la même bibliothèque. Pour chaque **procédure**, spécifiez les paramètres d'**entrée** ou de **sortie** (iconiques et de contrôle) requis, ainsi que leur documentation (**sémantique**).

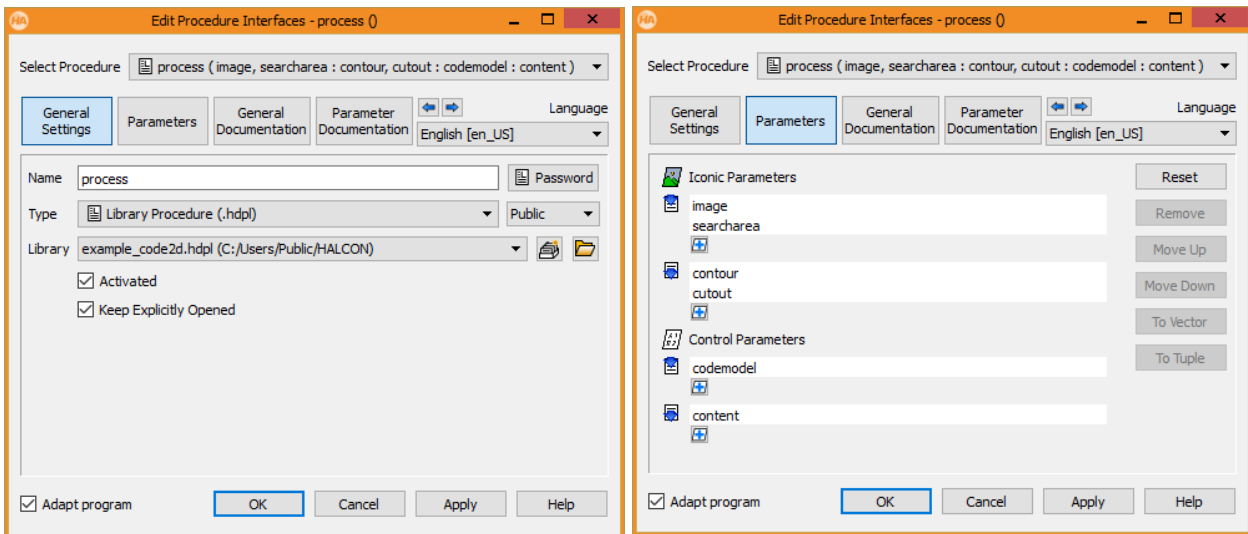


Figure 4 Des interfaces de procédure entièrement définies avec documentation des paramètres (sémantique) sont déterminantes.

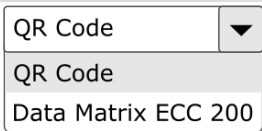



⚠ Important: Les « types de sémantique » corrects doivent être affectés aux paramètres d'entrée et de sortie des procédures HALCON dans l'onglet "Parameter Documentation" !

Pour l'exemple de traitement d'images de ce conseil technique (*example_code2d.hdp*), créez les paramètres de procédures suivants :

Procédure	Paramètre	Type	Sémantique
init	codetype	Entrée : paramètre de commande	chaîne (valeur : code QR, matrice de ECC 200)
	codemodel	Sortie : paramètre de commande	code 2D
process	image	Entrée : paramètre iconique	image
	searcharea	Entrée : paramètre iconique	zone
	codemodel	Entrée : paramètre de commande	code 2D
	contour	Sortie : paramètre iconique	xld
	cutout	Sortie : paramètre iconique	image
	content	Sortie : paramètre de commande	chaîne
finalize	codemodel	Entrée : paramètre de commande	code 2D
train	image	Entrée : paramètre iconique	image
	firstarea	Entrée : paramètre iconique	image
	codemodel	Entrée : paramètre de commande	code 2D
	content	Sortie : paramètre de commande	chaîne
	sizemin	Sortie : paramètre de commande	entier
	sizemax	Sortie : paramètre de commande	entier

Sémantique des paramètres

La sémantique des paramètres HALCON est particulièrement importante. Elle définit quels types de paramètres sont transférés vers les différentes procédures HALCON ou interrogés par l'application de vision. En fonction de la sémantique d'un paramètre HALCON, l'application de vision génère de manière dynamique un paramètre d'interaction approprié pour l'interface REST. Les paramètres d'interaction suivants sont générés dans l'exemple :

Paramètre de procédure HALCON	IDS NXT Paramètre d'interaction	
codetype		Liste déroulante Ctrl
image		Image
contour		Image avec marqueur
content	Code content: IDS-Techtip	Étiquette
searcharea		Zone réglable (section d'image)

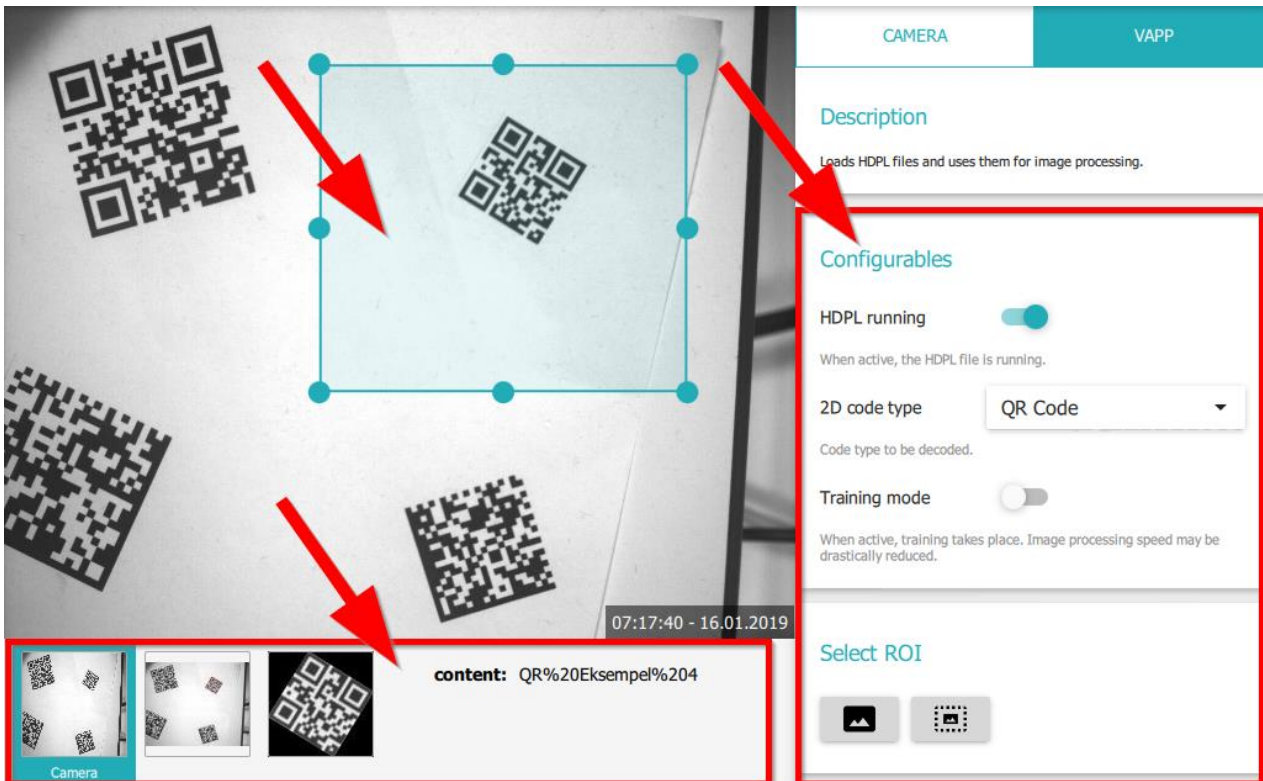


Figure 5 Les paramètres d'interaction sont affichés en tant qu'éléments graphiques de l'application de vision.

Traitement d'images

i Ce conseil technique décrit uniquement l'échange entre la « Dynamic Vision App for HALCON » et la bibliothèque HALCON avec les interfaces de procédure.

Dans la procédure « init », sont créés le paramètre de commande « codetype », pour la sélection du type de code, ainsi qu'un modèle de code 2D « codemodel », qui sert également de paramètre d'entrée dans d'autres procédures. Par ailleurs, un fichier contenant des données d'apprentissage est chargé ou créé afin de générer un modèle de code plus détaillé.

```

if (FileExistsTraining)
    read_data_code_2d_model(TrainingFilePath, codemodel)
[...]
else
    create_data_code_2d_model(codetype, [], [], codemodel)
    
```

Dans la procédure « process », les données d'image sont d'abord réduites à l'aide de la zone « searcharea » définie par l'utilisateur. HALCON recherche ensuite des codes 2D dans cette zone avec le « codemodel » (appris), puis transfère les résultats dans la chaîne « content » et leur position dans l'image en tant que XLD dans l'image « contour ». L'image « cutout » affiche le code découpé.

```
find_data_code_2d (image, contour, codemodel, [], [], ResultHandles, content)
[...]
reduce_domain(image, Region, cutout)
```

Dans la procédure « train », le modèle de code est encore amélioré et les données d'apprentissage sont enregistrées dans un fichier.

```
find_data_code_2d(ImageReduced, contour, codemodel, ['train'], ['all'],
ResultHandles, content)
[...]
write_data_code_2d_model(codemodel, TrainingFilePath)
```

La procédure « finalize » vide la mémoire et supprime le « codemodel ».

```
clear_data_code_2d_model(codemodel)
```

Téléchargement de la bibliothèque HALCON



Vous pouvez à présent utiliser la bibliothèque HALCON qui a été créée. Utilisez le bouton « Télécharger » (📁) de l'application de vision pour télécharger la bibliothèque HALCON préalablement créée. Un message vous informe lorsque le chargement de la bibliothèque s'est déroulé correctement. Dans ce cas, l'application de vision affiche les paramètres d'entrée et de sortie de la bibliothèque HALCON dans l'IDS NXT Cockpit. Le traitement d'images peut être utilisé immédiatement.

La bibliothèque HALCON reste en permanence en mémoire, même après le prochain redémarrage, jusqu'à ce qu'elle soit supprimée du système avec le bouton « Supprimer » (🗑️) ou remplacée avec le téléchargement d'une autre bibliothèque. Une seule bibliothèque peut être chargée et active à la fois. Vous pouvez cependant modifier le traitement d'images aussi souvent que vous le souhaitez.

Conseils

Performances

La combinaison de l'application de vision générique et d'une bibliothèque HALCON n'est pas optimisée en termes de vitesse et ne peut donc pas être comparé à la programmation d'une application de vision C++ spécialement mise en œuvre. L'exemple du conseil technique *example_code2d.hdpi* atteint à peu près 70 % de la vitesse de l'application de vision spécialisée du lecteur de code matriciel. Par contre, la « Dynamic Vision App for HALCON » présente l'avantage de permettre de tester le traitement d'images HALCON très rapidement et facilement sur une caméra basée sur une application de vision, sans vous obliger à écrire une seule ligne de code source C++ et à effectuer une compilation croisée pour la plate-forme de la caméra.

Gestion des exceptions

Si une exception non gérée se produit pendant l'exécution d'une procédure HALCON, l'application de vision redémarre et décharge la bibliothèque HALCON. Pour éviter un blocage, assurez-vous que toute erreur est traitée dans le cadre des procédures HALCON (gestion des exceptions avec try...catch).

Astuces

Éviter le redémarrage

Si l'application de vision se bloque pendant développement de la bibliothèque de traitement d'images HALCON, il peut être pertinent de ne pas redémarrer automatiquement l'application de vision à de multiples reprises. Vous pouvez empêcher cela avec une commande REST. Vous décidez ensuite manuellement quand l'application doit démarrer.

```
curl -u <user credentials> -X PUT  
/vapps/activated/dynamic_vision_app_for_halcon?temporary=true <device URL>
```

i Cette commande REST est valide une seule fois pour l'exécution actuelle de l'application de vision !

Affichage du journal

Pour afficher un journal détaillé de la « Dynamic Vision App for HALCON », par exemple, pour rechercher un dysfonctionnement de l'application de vision ou du traitement d'images, appelez l'URL suivante dans le navigateur :

```
http://<Geräte IP>/vapps/dynamic_vision_app_for_halcon/log
```

```
644-644 [13:31:17] . [dynamic_vision_app_for_halcon] 'Dynamic Vision App for HALCON' up and running.
644-644 [08:58:41] . [MyVApp] Loading engine...
644-644 [08:58:44] . [MyEngine] Updating library...
644-644 [08:58:45] . [MyEngine] Installing Configurables...
644-644 [08:58:45] . [MyEngine] Found Configurable codetype with semantic type "string" and default value "QR Code"
644-644 [08:58:45] . [MyEngine] Configurable codetype processed.
644-644 [08:58:45] . [MyEngine] Running init procedure for the first time...
644-644 [08:58:45] . [MyEngine] Initializing...
644-644 [08:58:45] . [MyEngine] Executing init procedure...
644-644 [08:58:45] . [MyEngine] Init procedure executed. Extracting data...
644-644 [08:58:45] . [MyEngine] Init procedure initialized parameter "codemodel"
644-644 [08:58:45] . [MyEngine] Installing ResultImages...
644-644 [08:58:45] . [MyEngine] Reading ResultImage "contour" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "contour" is "xld"
644-644 [08:58:45] . [MyEngine] ResultImage "contour" processed as Contour Image.
644-644 [08:58:45] . [MyEngine] Reading ResultImage "cutout" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "cutout" is "image"
644-644 [08:58:45] . [MyEngine] ResultImage "cutout" processed as HALCON Image.
644-644 [08:58:45] . [MyEngine] Installing ResultSources...
644-644 [08:58:45] . [MyEngine] Reading ResultSource "content" ...
644-644 [08:58:45] . [MyEngine] Semantic type of "content" is "string"
644-644 [08:58:45] . [MyEngine] Semantic type mapped to "String"
644-644 [08:58:45] . [MyEngine] ResultSource "content" processed.
644-644 [08:58:45] . [MyEngine] Installing Configurables...
644-644 [08:58:45] . [MyEngine] InputCtrlParameter "codemodel" is taken from initialization procedure.
644-644 [08:58:45] . [MyEngine] Installing ROIs
644-644 [08:58:45] . [MyEngine] Found IconicInputParam "image"
644-644 [08:58:45] . [MyEngine] Found IconicInputParam "searcharea"
644-644 [08:58:45] . [MyEngine] Found ROI "searcharea" with semantic type "region"
644-644 [08:58:45] . [MyEngine] Loaded procedures:
644-644 [08:58:45] . [MyEngine] Init: true
644-644 [08:58:45] . [MyEngine] Processing: true
644-644 [08:58:45] . [MyEngine] Training: true
644-644 [08:58:45] . [MyEngine] Finalize: true
```

Figure 6 Ce journal indique un processus sans erreur lors du chargement d'une bibliothèque HALCON.

La « Dynamic Vision App for HALCON » analyse les paramètres de procédure définis dans la bibliothèque et crée les éléments d'interaction de l'application de vision (voir marqueurs jaunes) tels que les éléments configurables, les entrées et les résultats. Les erreurs d'exécution dans le traitement d'images HALCON peuvent être identifiées et corrigées plus facilement avec ce journal d'application.

Si vous utilisez le fichier de sortie standard, les messages HALCON peuvent également être intégrés dans ce journal.

```
open_file('standard', 'output', stdout)
result := 'NIO'
fwrite_string (stdout, result + '\n')
```

Conclusion

La « Dynamic Vision App for HALCON » offre un moyen rapide et aisé pour exécuter le traitement d'images basé sur HALCON sur une caméra basée sur une application de vision, telle l'IDS NXT vegas. Cela vous évite d'avoir à programmer votre propre application de vision en C++, opération coûteuse en temps, et vous permet de vous concentrer sur la création des routines HALCON. Le mécanisme de téléchargement, simple, vous permet de changer la tâche de la caméra basée sur une application de vision IDS NXT aussi souvent que vous le souhaitez pendant l'exécution. Pas besoin de maîtriser le langage C++ ou la programmation des applications de vision. Traitement d'images intégré, « it's so easy ».

Si vous souhaitez en savoir plus sur le concept d'application de vision IDS NXT et sur son utilisation, lisez le [guide de développement des applications de vision IDS NXT](#) et regardez les [didacticiels vidéo IDS NXT](#). Pour une introduction à la programmation simple et intuitive de [votre propre application de vision machine HALCON](#), reportez-vous aux pages de MVTec consacrées aux produits HALCON.

D'autres astuces techniques et rapports d'application [sont disponibles sur le site Web d'IDS](#). Vous y trouverez également davantage d'informations sur [les nouvelles caméras industrielles basées sur une application de vision de la gamme IDS NXT](#).

Auteur

Dipl.-Ing. Heiko Seitz
Rédacteur technique

IDS Imaging Development Systems GmbH
Dimbacher Str. 6-8
74182 Obersulm
Allemagne

Tél. : +49 7134 96196-0
E-Mail : h.seitz@ids-imaging.de
Web : www.ids-imaging.com

© 2019 IDS Imaging Development Systems GmbH