

Szakdolgozat

Vámos Dániel

Debrecen
2008

Debreceni Egyetem

Természettudományi és Technológiai Kar
Szilárdtest Fizika Tanszék

**Átmenő furatos alkatrészek optikai
vizsgálata**

Témavezető:

Dr. Cserháti Csaba
egyetemi adjunktus

Készítette:

Vámos Dániel

mérnök informatikus

Debrecen
2008

1. Bevezetés :	4
2. A fejlesztőkörnyezet bemutatása	6
2.1 Labview.....	6
• Előlap panel.....	6
• Blokk-diagramm.....	7
2.2 Labview Vision.....	8
• Az IMAQ Create és IMAQ Dispose függvények	8
• Korreláció	10
• Mintaillesztés (Pattern Matching).....	11
• Az IMAQ Match Pattern 2 függvény	12
• IMAQ Setup Learn Pattern 2:	13
• IMAQ Learn Pattern 2 :	14
• IMAQ Setup Match Pattern 2:	14
• Geometric Matching	15
• IMAQ Setup Match Geometric Pattern:.....	17
• IMAQ Match Geometric Pattern:.....	17
• IMAQ Overlay függvények	18
• Draw Pattern Matching Position VI:.....	19
• IMAQ Coordinate System (Koordináta rendszerek és használatuk).....	20
• IMAQ Find CoordSys (Rectangle)	21
• IMAQ Find CoordSys (2 Rectangle)	21
• IMAQ Find CoordSys (Pattern)	22
3. Optikai vizsgálat	23
• A vizsgálóállomás eszközeinek felsorolása:	23
• A mérőállomás összeállítása:	24
• Látómező.....	25
• Szenzor.....	25
• A lencse.....	25
• Hibák és Korrigálásuk.....	26
4. Program:	28
• Gépigény (ajánlott minimum konfiguráció):.....	29
• Futtató Környezet:.....	29
• Könyvtárszerkezet Létrehozása és Kötelező beállítások.....	29
• Programegységek	31
• Template_Creator(2in1.VI).....	31
• Config_Deafault_Match_Mode	33
• Select Templates to Investigate.....	36
• Search Selected Templates.....	38
• Show Results.....	41
• Munkamenet:	44
5. Algoritmus:	44
Algoritmus ábra	45
6. Összefoglalás:	46
7. Irodalomjegyzék:	47
8. Köszönetnyilvánítás:	48

1. Bevezetés :

A projekt célja egy olyan alacsony költségű optikai vizsgálóállomás kifejlesztése, amely a National Instruments hardware és software megoldásain alapulva az elektronikai panelgyártás illetve általában a termékgyártás végellenőrzési folyamatát támogatja. A projekt során kifejlesztendő megoldás olyan rugalmas megközelítést alkalmaz az optikai vizsgálati technikában, amely egy új piaci szegmens kialakulását célozza meg, hatékony és gazdaságos megoldást kínálva a nagy változékonyságú termékeket előállító cégek számára is.

A képfeldolgozás gyorsan fejlődő terület, amelyet nem utolsó sorban a digitális fényképezés elterjedése is segít. A nagy gyártók korábbi befektetéseiket nehezebben tudják hozzáigazítani a gyorsan fejlődő hardware és software lehetőségekhez. A projekt célja egy olyan általános célra kifejlesztett olcsó, hardware és software elemek kombinálásával, egy a termékvizsgálatra specializált optikai vizsgálóállomás megvalósítása. Egy ilyen megoldás hiányában a gyár jelenleg nem alkalmazza az anyavállalat által kidolgozott ipari képfeldolgozási technológiát. A projekt keretében kialakításra kerülő vizsgálóállomás versenyképes megoldás lehet más elektronikai gyártó cégek számára is, melyekből több is települt Magyarországra.

Egy olyan iparban, mint a nyomtatott áramkör gyártás, a mindig növekvő minőséget célzó vevői igények és a nemzetközi fejlődés nap mint nap kihívást támaszt a gyártó cégek számára.

A termékminőség biztosításának a legjobb módja az automatikus optikai ellenőrzés (AOI). Ez a program ideális kis sorozatban gyártott és egyedi termékek, valamint prototípusok ellenőrzésére, továbbá új termékek nagysorozatú gyártásának előkészítésére.

Az automatizált optikai vizsgálatok szerepe több okból is növekszik az elektronikai iparban. A hagyományos szemrevételezési vizsgálatokkal nagyon sok hiba kiszűrhetővé válik. A jelenleg használatos automatizált képfeldolgozási technikák nem vetekszenek az emberi látás intelligenciájával, ezért az operátor által végzett vizuális ellenőrzést nem kiiktatva, azt hatékonyan képesek támogatni.

Az alkatrész méretek csökkenése, a kis darabszámú, de változatos termékpaletta, folyamatosan változó gyártási technológiák, valamint az eszközök bonyolultságának növekedése miatt elengedhetetlen a vizuális vizsgálat folyamatainak automatizálása.

A diplomamunkám egy egyetemi kutatóhely és egy regionális gyártó együttműködésére épül. A National Instruments debreceni gyára a Debreceni Egyetemmel megalapítása óta intenzív kapcsolatban áll. A gyár debreceni telepítésével a városba került egy jelentős technológia kultúra is, amit a cég mérés-technikai és képfeldolgozási termékei képviselnek. Ezeket az eszközöket ingyenes licenc szerződéssel bocsátotta az egyetem rendelkezésére az anyavállalat, és több esetben sor került hardware eszközök ajándékozására is. Az eszközök oktatási alkalmazás révén egyre több hallgató és oktató ismerkedett meg ezekkel a technológiákkal és alkalmazta azokat munkája során. Jelen pályázat ezen a kétoldalú együttműködésen alapulva a debreceni egyetem szellemi kapacitását és a gyár által kínált technológia környezetet kívánja a fejlesztés során kamatoztatni. A projekt sikeres megvalósulása a gyártáson túlmutatóan segítheti a fejlesztési kapacitások részleges Debrecenbe telepítését is.

Az ipari képfeldolgozó rendszerek piacán kétféle tendencia figyelhető meg. A célfeladatra készült magas költségű megoldásokat elsősorban a forrasztás minőségének ellenőrzésére alkalmazzák. Ezen a piaci területen néhány gyártó nagy fejlesztési háttérrel kínál komplett megoldásokat. A másik terület az egyedi, konfigurálható megoldások piaca, ahol egy konkrét, néhány lépésből álló vizsgálatra a software és hardware elemekből a felhasználó igényei szerint konfigurálható vizsgálatokra van lehetőség. A National Instruments ezen a területen mind hardware (képgyűjtő kártyák, Compact Vision), mind software (Labview Imaq Vision, Vision Builder) megoldásokkal jelen van.

A vizsgálóállomás kifejlesztése során elsősorban azokra a hatékony képfeldolgozási algoritmusokra támaszkodhatunk, amely az IMAQ Vision software részét képezik. A tudományos hozzáadott értéket elsősorban ennek a technológiát hatékony adatbázis-kezelési technikákkal és a digitális képrögzítés legújabb olcsó technológiáival való integrálása jelenti, amit a képminőség vizsgálatára és mérésére alkalmazandó új technológia megoldások tesznek lehetővé.

2. A fejlesztőkörnyezet bemutatása

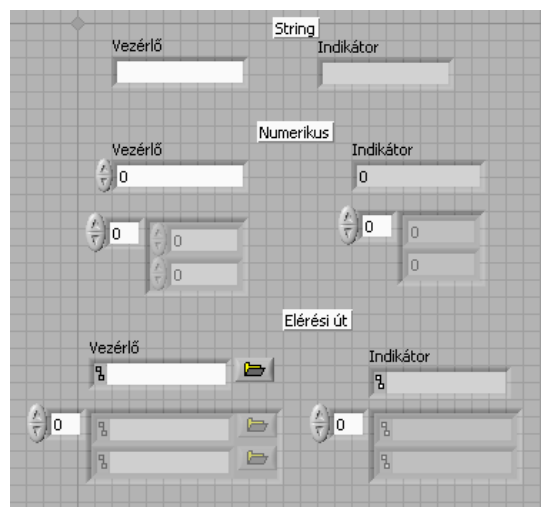
2.1 Labview

Labview (Laboratory Virtual Instrumentation Engineering Workbench) maga egy platform és egyben egy grafikus fejlesztői környezet is egyben, amit maga a National Instruments fejlesztett ki [1]. A grafikai nyelvet „G”-nek nevezik, ami egy grafikus adatfolyam .programozási nyelv Eredetileg az Apple Macintosh-ra készült 1986-ban, azóta számos platformra kifejlesztették (Windows, Linux, Unix). Jelenleg a legfrissebb verziója a 8.6 amit 2008 augusztusban adtak ki.

A Labview-ban való fejlesztés alatt az előlapi panel és a blokk diagramm szerkesztését értjük.

- **Előlapi panel**

Ez a komponens van közvetlen kapcsolatban a felhasználóval, hisz ezen keresztül manipulálható közvetlenül a program. Itt adhatunk úgynevezett bemenő paramétereket a programnak. Ezen attribútumokkal, vagy azokon operál a blokk diagramm, majd bizonyos műveletek elvégzése során szintén az előlapi panelre továbbítja az adatot a felhasználó számára. Az előlapi eszközöket két csoportra lehet bontani. Vezérlőkre és kijelzőkre (Control and Indicator). Attól függően, hogy milyen típusú adatot szeretnék manipulálni vagy megjeleníteni, úgy változik a kontrol és az indikátor is. Az adat belső reprezentációjától függően a Labview automatikusan felismeri, hogy milyen típusú adatot szeretnék megjeleníteni. Tömb adatszerkezet esetén is megtörténik az automatikus típuskonverzió.



- **Blokk-diagramm**

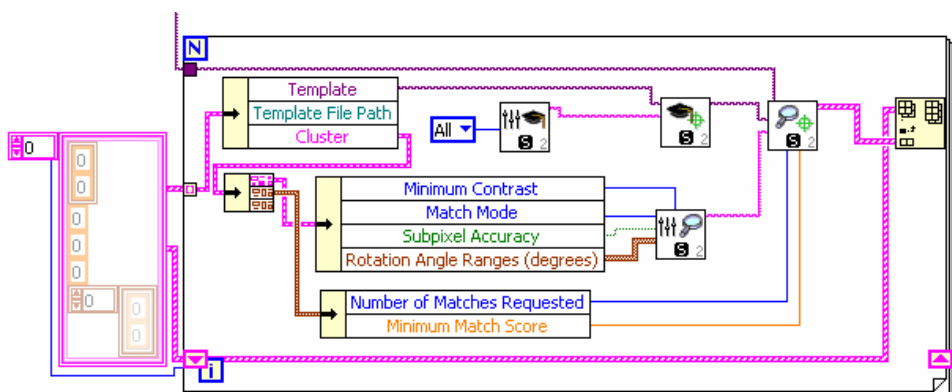
Tágabb értelemben ez maga a fejlesztői környezet. Míg az előlapi panel a külső megjelenésért felelős, a blokk-diagramm az adatfolyam programozásért felel. Lényegében a huzalok segítségével teremtünk kapcsolatot a csomópontokkal. A csomópontok funkciójuk szerint lehetnek függvények, struktúrák és SubVI-ok (a hagyományos programozásban a szubrutinnak megfelelő szerkezet). SubVI-ok felhasználhatósága igen széleskörű. A fejlesztő saját maga definiálhat egy SubVI-ban függvényeket, melyek tetszőleges mélységben egymásba ágyazhatók. Az ismétlődő kódrészlet kiemelése mellett a rekurzió is könnyebben megvalósítható. Mindezek mellett javul a program blokk diagrammbeli struktúrájának az áttekinthetősége.

A program végrehajtását is ez a diagramm határozza meg, ami az adatáramlás sorrendjében hajtódik végre.

Programozás során az egyik leggyakrabban használt adatszerkezet a „lista”. Erre az adatszerkezetre bizonyos feltételeket állítva már a „sor” adatszerkezethez jutunk. A Labview tömb szintjén is meg tudja valósítani mindkettőt a beépített függvényei segítségével. Véleményem szerint a leggyakrabban használatos alap adatszerkezeteket elő lehet állítani Labview-ban, igaz nem mindig egyszerű a dolgunk.

A következő képen a blokk-diagramm egy részlete látható:

Egy összetett struktúrát tartalmazó konstans, cluster műveletek, mintaillesztő VI-ok és tömb függvények.



A ciklusok és más programozási struktúrák közül a leghasznosabbnak bizonyultak számomra az egymásba ágyazott, a lap és az eseményvezérlés struktúra. Ezek segítségével könnyen szabályozható az adatáramlás (data flow). Míg az első kettő az egymás-utániságot garantálja, a harmadikat egy általunk kiválasztott esemény aktiválja. Processzor terhelés szempontjából az eseményvezérlés struktúra a leggazdaságosabb, de a jelenlegi számítógépes konfigurációk mellett nincs számottevő különbség.

2.2 Labview Vision

A képfeldolgozási feladatok megoldásához a Labview hagyományos jelfeldolgozási rendszerét kiterjesztették hardware és software oldalon.

Hardware oldalon a megfelelő driverek és csatlókártyák segítségével lehetőség van analóg, USB, firewire és kameranál csatlakozású kamerák jeleinek fogadására. A speciális képfeldolgozást biztosító függvények a Labview- on belül az IMAQ programcsomagban találhatóak meg. A képfeldolgozási feladatok interaktív konfigurálását az IMAQ részét is képező Vision Assistant program segíti. Kifejezetten a gyártási - minőségellenőrzési feladatok interaktív programozását segíti a Vision Builder program, amely használja a Vision Assistant szolgáltatásait is. Ezekkel, a varázslókkal összeállított képfeldolgozási megoldásokból Labview program generálható, amely így könnyen továbbfejleszhető [2].

A Labview egy kiegészítő modulja a Vision, ami a képfeldolgozással kapcsolatos függvényeket tartalmaz. Az általam készített program számos ilyen függvényt használ, így szükségét érzem néhány ezen beépített függvények közül néhányat ismertetsek.

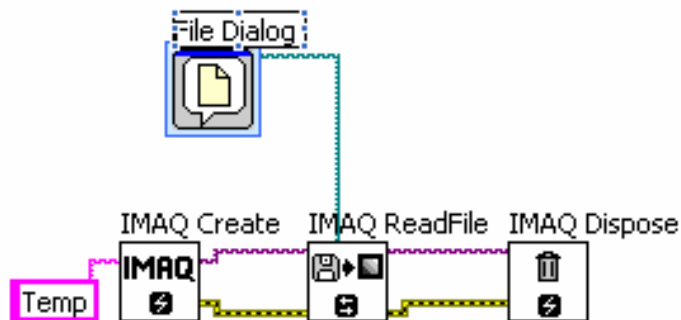
- **Az IMAQ Create és IMAQ Dispose függvények**

Mikor képfájlt akarunk beolvasni a legfontosabb lépésünk egy IMAQ Create függvény hívása. Ez egy ideiglenes memória helyet foglal le a memóriában, melynek egy egyedi nevet kell adni. Ez egy kötelező paramétere az IMAQ Create függvénynek. A létrehozás maga nem egy veszélytelen művelet, hisz ez memóriát foglal le amit fel is kell szabadítani. Ezért nagyon fontos meghívni minden egyes IMAQ Create után az IMAQ Dispose függvényt [2,3].

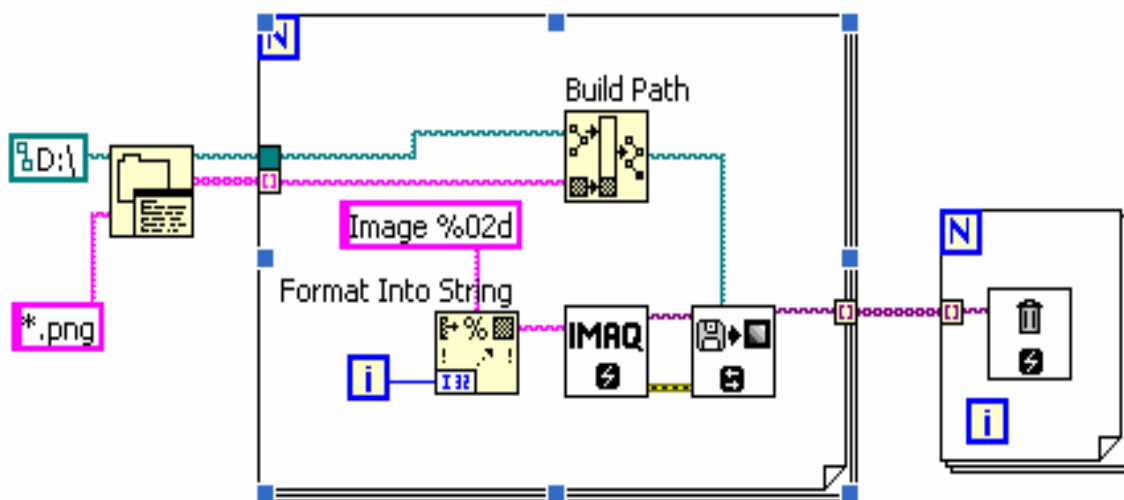
Ez törli a képet és felszabadítja a lefoglalt memóriarészt. Természetesen akkor használatos ez a függvény mikor a már nincs szükségünk a képre további processzálás végett.

Figyelnünk kell arra is, hogy mikor egy Labview alkalmazást megszakítunk, az IMAQ Create függvény által lefoglalt terület a memóriában marad.

A következő képen egy kép helyes beolvasását láthatjuk, majd a lefoglalt terület felszabadítása látható.



Fejlesztéseink során gyakran találkozunk olyan esettel mikor egy könyvtárból nem csak egy darab képfájlt szeretnénk beolvasítani hanem magát az egész könyvtár tartalmát. Tudjuk viszont, hogy minden egyes létrehozandó memóriaterületnek egyedi nevet kell adni. Ilyenkor tanácsos a következőképp eljárni.



Az elérési utat megadjuk a List Folder függvénynek, ami minden egyes fájl nevet továbbít a ciklus felé. A ciklus egyesével leválogatja a képek neveit, előállítja azokat, majd az IMAQ Read File függvénnyel egymás után meg is nyitja azokat.

Az egyedi neveket a ciklusváltozó és a sztring konstans konkatenációja eredményez.

A ciklus a képekből tömböt épít és továbbítható az összes beolvasott kép a processzáls felé.

A képek törlését is ciklus segítségével valósítom meg.

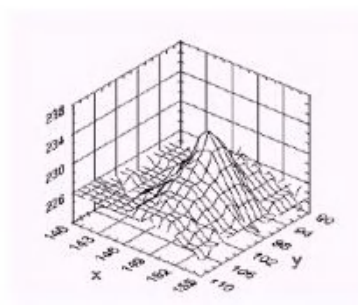
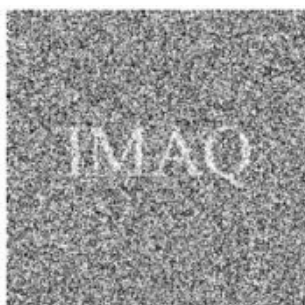
- **Korreláció**

A korrelációt akkor használjuk, ha képek közötti hasonlóságot keresünk, vagy a képen egy bizonyos tartomány pontos pozíciójára vagyunk kíváncsiak, esetleg két kép közötti eltolódást, annak irányát és nagyságát kell meghatározni [2,5,6,7,8]. Egy $w \times h$ méretű k kernelt alkalmazva $[(mind\ w, mind\ h\ páratlan\ és\ m=(w-1)/2, ill.\ n=(h-1)/2\ a\ fél\ szélesség, illetve\ fél\ magasság)]$ az f képre, a korrelációs művelet eredménye a következőképp számítható ki:

$$g(x,y) = \sum_{j=-n}^n \sum_{i=-m}^m k(i,j) f(x+i,y+j)$$

vagyis a k kernel elemei (a konvolúcióval ellentétben) a mártixelem alatt lévő pixel intenzitásértékeivel szorzódnak. Mivel a művelet a világos részek fényességét még jobban megnövelik, ezért a fenti kifejezés normált változatát használjuk:

$$g'(x,y) = \frac{\sum_{j=-n}^n \sum_{i=-m}^m k(i,j) f(x+i,y+j)}{\sum_{j=-n}^n \sum_{i=-m}^m f(x+i,y+j)}$$



VI.13. ábra A legjobb illeszkedés koordinátái (149,99).A VI. 13. ábrán a középén látható mintázatot kerestük a bal oldali zajos képen. A jobboldalon látható a korrelációs felület, mutatva a legjobb illeszkedés helyét (149,99).

- **Mintaillesztés (Pattern Matching)**

Mintaillesztés során, egy szürkeárnyalatos képen azt a terület határozzuk meg ami a paraméter templéttel, egy referenciakép előre meghatározott részével egyezett meg.

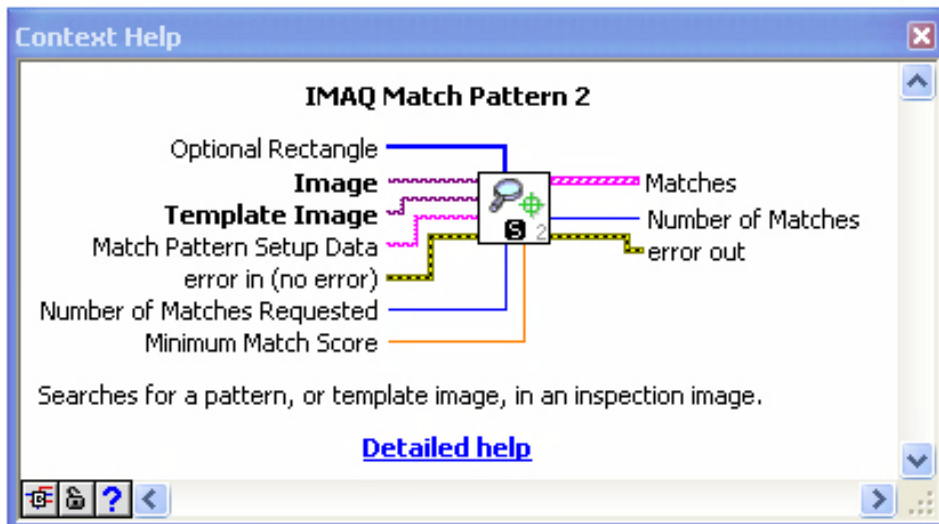
Maga a minta egy idealizált reprezentációja a referencia kép részének.

Mikor mintaillesztéssel dolgozunk, keresünk, akkor egy mintát, templétet kell készíteni arról az objektumról, amit szeretnénk megkeresni. A gépi látás alkalmazás megkeresi a minta minden egyes előfordulását a forrás képen, melyet egy illesztési pontszámmal lát el. Ez az érték mutatja meg az illeszkedés jóságát.

Mintaillesztés során, a program a templét minden egyes előfordulását felkutatja a forrás képen, függetlenül a fényviszonyoktól, geometriai transzformációktól (eltolás, forgatás, és a templét méretarányaiban való különbsége) a forráson való előforduláshoz képest [4,6,7,8].

A mintaillesztő algoritmus, ami minden esetben a korrelációs függvény számításán alapszik [2], az egyik legfontosabb függvény a gépi látásban, mert számos terület ezen függvényen alapul. Az algoritmus főbb felhasználási területei:

- Csoportosítás, Osztályozás: Meghatározza egy kiválasztott alakzatnak a pozícióját, ezáltal eltéréseket, elmozdulásokat tudunk vizsgálni.
- Mérés: Távolság, átmérő, forgásszögek mérésére alkalmas. Lehetőség van az objektumok méret szerinti osztályozására is.
- Vizsgálatok: Egyszerű hibák, hiányzó alkatrészek felkutatása vagy egyszerűen egy olvashatatlan nyomtatvány felismerése.



- **Az IMAQ Match Pattern 2 függvény**

Bemenő paraméterek:

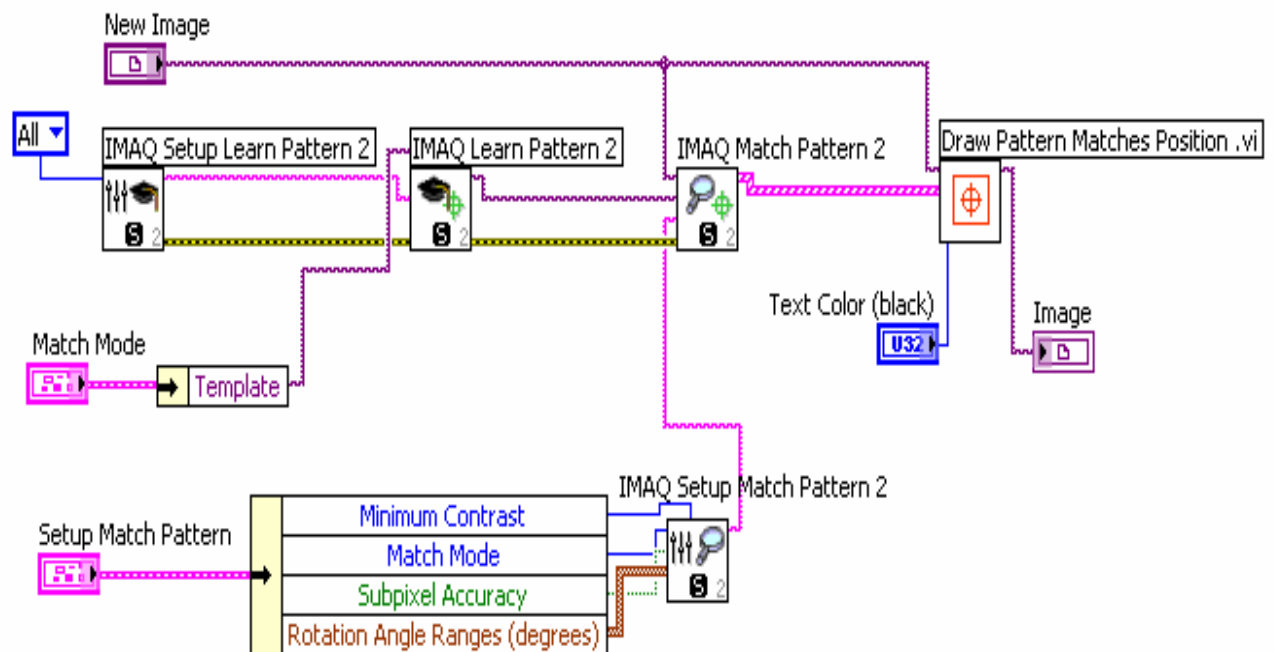
- **Optional Rectangle:** Négyelemű tömb , ami a feldolgozandó területet határozza meg. Ezek sorrendben: bal , fel, jobbra, le. Ha nem adjuk meg ezt a paramétert, akkor a keresés a forrás kép egészére kiterjed.
- **Image:** A vizsgált kép, amin a keresést végre akarjuk hajtani, amin a *mintát* keressük.
- **Template Image:** Az a minta, amit a vizsgálandó képen keressünk.
- **Match Pattern Setup Data:** Sztring paraméter, ami információkat szállít az IMAQ Setup Match Pattern 2 VI tól, vagy az IMAQ Advanced Setup Match Pattern 2 VI felől. Ha nem adjuk meg ezt a paramétert akkor a VI az alapértelmezett paraméterekkel látja el az összes beállítást, ami egy üres sztring konstans.
- **Number of Matches Requested:** Az a számérték , ami megmondja ,hogy mennyi az általunk elvárt találatok száma.
- **Minimum Match Score:** Ezzel szabályozható a keresés pontossága. Az intervallum 0 -1000 –ig tart. Magasabb számérték, pontosabb egyezést eredményez.

Kimenő paraméterek:

- **Matches:** Találatok koordinátái.
- **Number of Matches:** Találatok száma.

Önmagában ez a VI sem oldja meg a mintaillesztést, hisz számos paraméter szükséges ahhoz, hogy akár egy egyszerű keresést is el tudjunk végezni. Match Pattern Setup Data helyes paraméterezése nélkülözhetetlen.

A következő kódrészlet bemutatja a Mintaillesztés egy lehetséges alkalmazását.



A kódrészleten látható függvények:

- **IMAQ Setup Learn Pattern 2:**

A templét mintaillesztési beállításai adhatók meg ennek a VI-nak a segítségével.

Bemenő paraméterek:

- All: A templét forgatott és eltolt variációinak keresése .
- Shift Information : A templét eltolt variációinak keresése.
- Rotate information : A templét forgatott variációinak keresése.

Kimenő paraméterek:

- Learn Pattern Setup Data: A mintaillesztéshez szükséges adatok.

- **IMAQ Learn Pattern 2 :**

Egy leírást készít a templét képről, amit a mintaillesztés során használni fogunk. Ez a leíró attribútum a paraméterhez kapcsolódik a mintaillesztés során és információkat szolgáltat a kereső algoritmusnak az illesztéshez.

A leíró attribútum a következőképp áll össze:

IMAQ Setup Learn Pattern 2	+	IMAQ Learn Pattern 2	=	Teljes leíró attribútum
+ Learn Mode		+ alapértelmezett attribútum		

- **IMAQ Setup Match Pattern 2:**

Ez a VI Explicit mintaillesztési beállításokért felel. Előnye, hogy minden egyes templétet egyéni mintaillesztési konfigurációval láthatunk el.

IMAQ Match Pattern 2 VI kimenetén egy cluster – tömböt kapunk, ami a talált egyezések adatait tartalmazza. A tömb elemszáma megegyezik , az vizsgálandó képen megtalált objektumok számával.

Draw Pattern Matches Position.vi – t az IMAQ Overlay témakörben részletezem.

Használata/Személyes tapasztalat:

Ez a fajta mintaillesztési eljárás kimondottan alkalmas olyan képek esetén, ahol az egységnyi területre eső pixelértékek magasak. Olyan képek vizsgálatára alkalmas ahol nagyon változatos a színösszetétel.

A program írásakor azért döntöttem ezen fajta mintaillesztési módszer mellett, mert a nyomtatott áramköri lapkán az átmenő furatos, viszonylag nagyméretű alkatrészek meglétét kellett leellenőrizni. A nagy méret miatt, az alkatrészekben a fényviszonyok sokat változtak, ami lehetetlenné tette a geometriai illesztést. Ez ugyan lassította a futási időt, viszont javult a hatékonyság.

- **Geometric Matching**

Mintaillesztési eljárások második variációja, mikor a minta geometriai tulajdonságait vesszük figyelembe.(pl.: szélek)

Geometriai illesztés során, a szürkeárnyaltos képen a paraméter templéttel megegyező területet határozzuk meg.

Lényeges különbség a Pattern Matching és ezen illesztési módszer között, hogy míg a másik eljárás a pixel értékeket veszi figyelembe, ez az eljárás már egyedi mintákkal dolgozik. Egy geometriai illesztésre kiválasztott templétnek számos egyedi attribútummal kell rendelkeznie, nem lehet egy akármilyen képet választani.

Geometriai illesztés akkor javasolt, mikor a mintán és a vizsgálandó felületen a geometriai sajátosságok megfelelően elkülönülnek. (pl.: fekete alapon, fehér eszközöket)

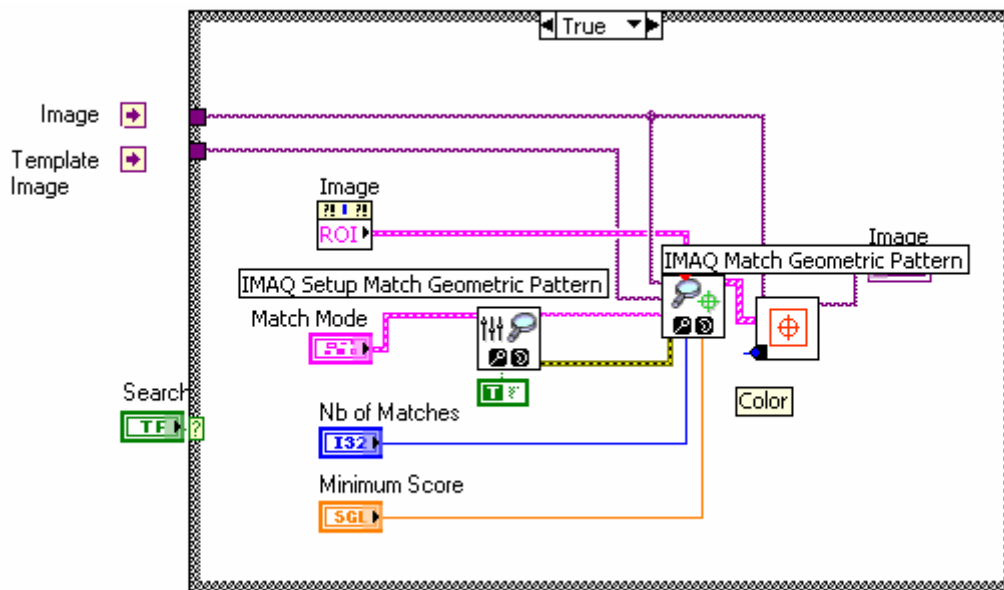
A Vision csomag rendelkezik egy templétek előállítására alkalmas programmal, amit sajnos nagyon elrejtett a felhasználók elől és hosszadalmas keresés után lehet csak megtalálni. Általában ezen az elérési úton lehet eljutni :

C:\Documents and Settings\All Users\Start Menu\Programs\National Instruments\Vision

Itt találjuk a Template Editor alkalmazást, aminek segítségével mintákat lehet létrehozni, mindkét fajta mintaillesztési algoritmushoz.

Geometriai illesztéshez, készítsünk el egy mintát, amit majd használni fogunk az illesztésnél. Kiválasztjuk a File ->New Template és azon belül Geometric Matching Template opciót, és egy képfájlt.

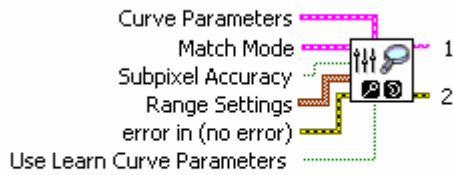
A Select Template Region fülön az alkalmazás közvetlenül egy ROI függvénnyel kezd. (Region of Interest). Kivágjuk a mintának való alakzatot és tovább lépünk a Define Cuves fülre, ahol az Edge Detection függvény fut, ami automatikusan megkeresi az éleket. Természetesen ezek felüldefiniálhatóak, és a felhasználó kedvére rajzolhat be kívánt széleket. A harmadik fülön ezek a beállítások finomíthatóak tovább. Az utolsó fülön a Match Offset –et helyezhetjük el. Ennek megfelelő elhelyezése gyorsíthatja meg a keresési eljárást. Tanácsos a minta egyedi pozíciójára elhelyezni, lehetőleg annak is a közepére. Az így elkészült mintával már tud dolgozni a Geometric Matchig függvény.



Részlet egy Geometric Matching függvényt használó programból

Szemmel látható, hogy a forrás sokkal kisebb, mint a Pattern Matching –et alkalmazó módszer. Ennek oka, hogy legtöbb illesztési beállítást Template Editorral végeztünk el. Minden geometriai illesztéshez szükséges adatot hozzáfűzött a mintánkhoz.

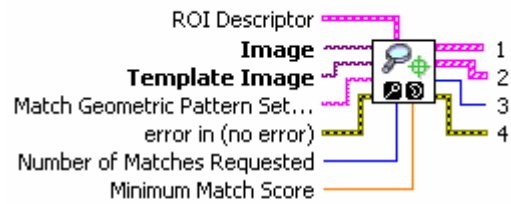
IMAQ Setup Match Geometric Pattern



Outputs

- 1 Match Geometric Pattern Set...
- 2 error out

IMAQ Match Geometric Pattern



Outputs

- 1 Matches (pixel)
- 2 Matches (real-world)
- 3 Number of Matches
- 4 error out

A két függvény fontosabb paramétere:

- **IMAQ Setup Match Geometric Pattern:**

- **Curve Parameters:** értéket akkor tanácsos adni, ha nem a Template Editorral készített mintát használjuk
- **Range Settings:** A keresési időt redukáló illesztési konstansokat tartalmaz.
 - **Rotation Angel:** Az illesztés során használható elforgatás mértéke.
 - **Sacle Factor:** Skalárfaktor.
 - **Occlusion:** Az illesztés százaléka. Ahhoz, hogy egy alakzatra illeszkedjen a geometriai mintánk akkor ennek az értéknek 50%-osnak kell lennie. Ez az alapértelmezett beállítás. Ha mi magunk adunk meg értéket vigyáznunk kell, a túl alacsony érték megadásával, mert az illesztés jóságát veszélyeztethetjük.

- **IMAQ Match Geometric Pattern:**

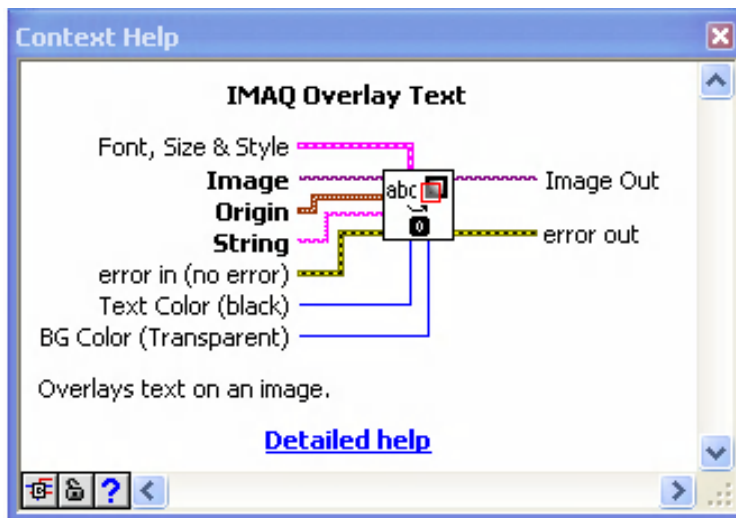
- **ROI Descriptor:** A vizsgálandó képnek azon része, amin az előfordulásokat keressük. Ez a rész egy téglalap vagy annak elforgatása lehet.

- **IMAQ Overlay függvények**


Ezen beépített függvények segítségével valósítható meg a kijelzőre való írásos szöveg felvitele, alakzatok rajzolása, amire szükségünk lehet bizonyos alakzatok címmel való ellátása, keresett objektumok bekeretezése a könnyebb észrevehetőség végett. Távolság, terület mérés során a mért értékeket feljegyezhetjük magára a képre, hisz gyakran gondot okozna egy külön táblázatból az adott objektumhoz tartozó érték kikeresése.


Az IMAQ Overlay függvénycsalád segítségével geometriai alakzatokat és szöveget helyezhetünk el a képen. Természetesen ezen alakzatok színét a fejlesztő adhatja meg [2,3].

IMAQ Overlay Text az egyik leghasznosabb beépített függvényünk. Lekérve a szöveges segítséget a VI-ról (Context help) láthatjuk milyen kötelező paraméterei vannak a függvénynek.



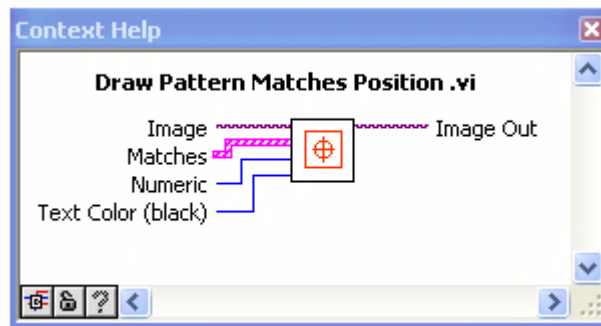
- **Image:** A forrás kép, ami a műveletet végezzük.
- **Origin:** A forrás képen lévő pozíció, ahova szeretnénk a kiíratást végezni, ez a paraméter nem más, mint a vízszintes és függőleges pozíciók értékei egy clusterben tárolva.

 X is the x-coordinate.

 Y is the y-coordinate.

- **String** : Az a karaktersorozat, amit ki szeretnénk írni.
- **Draw Pattern Matching Position VI:**

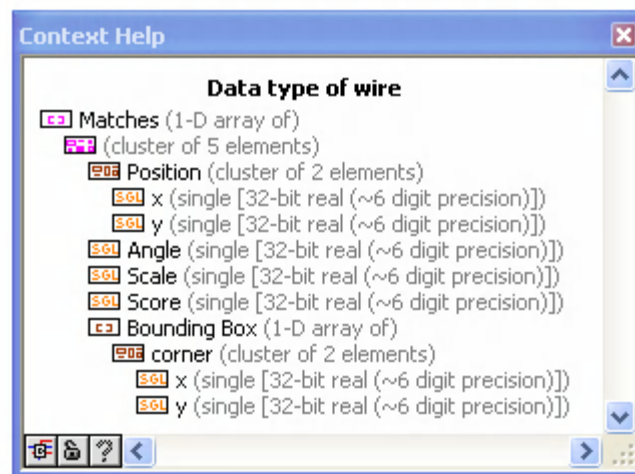
Ennek a VI-nak a segítségével a kijelzőre kirajzolhatjuk a mintaillesztés eredményét és a megtalálás sorrendjét.



Paraméterei:

1. Image: Az a kép, amire akarunk rajzolni, amire a kiíratást akarjuk végezni.
2. Matches: A IMAQ Match Geometric Pattern és az IMAQ Match Pattern 2 kimenete köthető rá. Az illesztés során egy minta többszöri előfordulása is szerepelhet a vizsgálandó képen, ezért ennek a paramétere egy tömb, ami minden egyes találatról információkat hordoz.
3. Numeric: Ide egy numerikus konstans állítható be, a kezdeti sorszám állítható be.
4. Text Color: A kiírt szöveg színe.

A Matches Paraméter értékkomponensei :



Ezeket a bemenő paramétereket az illesztés során a megtalált alkatrészek Match Offset koordinátaiból határoztam meg, ezen értékek alapján állítottam elő az objektumokat jelölő téglalapokat is.

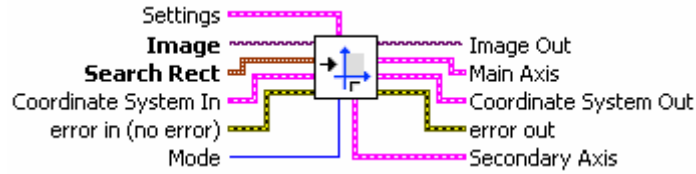
- **IMAQ Coordinate System (Koordináta rendszerek és használatuk)**

Koordináta rendszer VI-okat előszeretettel használhatjuk arra, hogy különböző helyen koordináta tengelyeket definiálhassunk a képen. Ezek segítségével mintaillesztési élkeresési feladatokat végezhetünk el [2,3].

Három beépített VI áll rendelkezésünkre:

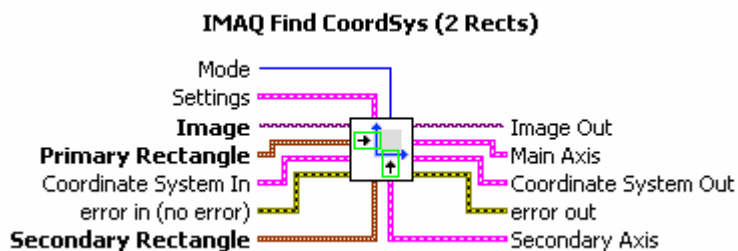
1. **IMAQ Find CoordSys (Rectangle)**
2. **IMAQ Find CoordSys (2 Rectangle)**
3. **IMAQ Find CoordSys (Pattern)**

- **IMAQ Find CoordSys (Rectangle)**



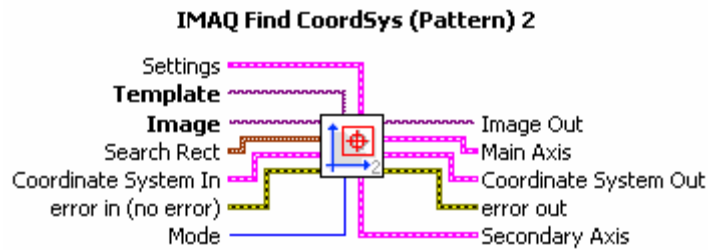
Egy koordináta rendszert készít a Search Rectangle paraméteret használva. Először egy ROI-t hozunk létre, amely tartalmazza az objektum két nem párhuzamos élét. A tartomány méretét úgy kell megválasztani, az összes vizsgálni kívánt képe tartamazza ezeket az éleket. Ezután a koordináta rendszert a VI élkereséssel határozza meg. A függvény ezen kívül kirajzolhatja a keresési területet, ahol a keresést kívánjuk végrehajtani, a kereső vonalakat és a széleket is kirajzolhatja.(Settings paraméter)

- **IMAQ Find CoordSys (2 Rectangle)**



A koordináta rendszer létrehozás szintén él-detektálással történik. A lényeges különbség az előző VI-hoz képest, hogy két ROI paramétert vár a függvényünk, és azokon a területeken keres koordináta tengelyhez való éleket.

- **IMAQ Find CoordSys (Pattern)**



Koordináta tengelyek létrehozásának egyik legegyszerűbb módja. Nincs másra szükségünk, mint egy mintaképre, ami tudjuk, hogy szerepel az eredeti képen. A függvény ezt a mintát felhasználva hoz létre egy koordináta tengelyt az egyezés helyén. Lényegében ez a függvény is mintaillesztést hajt végre a kép egészén. Gyorsabb programot készíthetünk, ha a kereséseket ilyen koordináta tengelyekkel határolt részekben tesszük.

3. Optikai vizsgálat

- A vizsgálóállomás eszközeinek felsorolása:

ASUS GX1105N Switch több kamera egyidejű használatához

Kamerák:(1)

- Blaser scA 1000 – 30GC
- S/N: 20826219
- Képméret: 1032 x 778
- Vízszintes és Függőleges felbontás: 96 dpi
- Bitmélység: 32 bit

(2)

- Blaser asA 1600 – 14 GC
- S/N: 20826644
- Képméret: 1624 – 1234
- Vízszintes és Függőleges felbontás: 96 dpi
- Bitmélység : 32 bit

Megvilágító rendszer elemei:

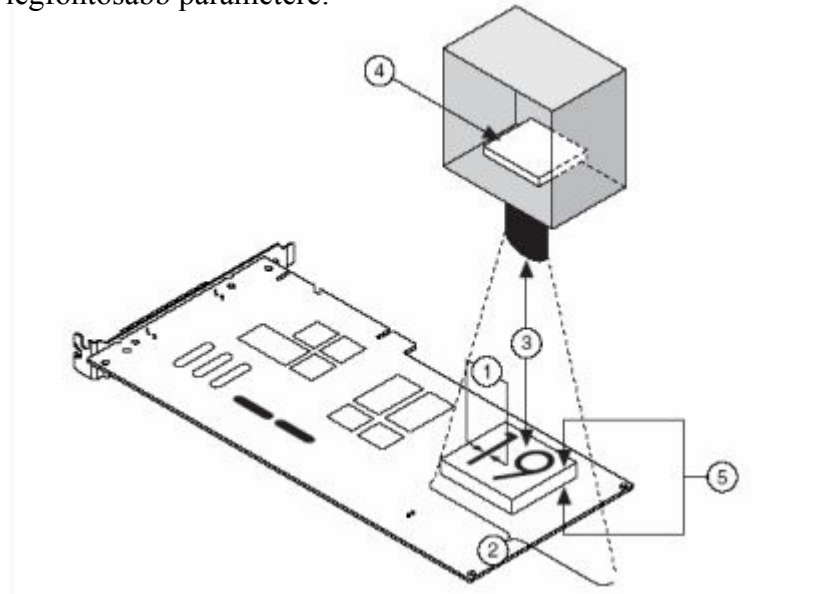
- Luxeon K2 star LED
- Lencse 10° / 40°
- Vezérelt LED driver

- A mérőállomás összeállítása:

A megfelelő minőségű és felbontású kép elkészítéséhez elengedhetetlen a kamera helyes távolságának, fókuszának beállítása a mérendő felülethez képest. [4,5]

A mérőállomás összeállításának öt legfontosabb paramétere:

1. felbontás
2. látómező mérete
3. munkatávolság
4. szenzorméret
5. mélységélesség



- Felbontás

Kétfajta felbontásról beszélhetünk, a pixelfelbontásról és a térbeli felbontásról. A szükséges pixelfelbontást a legkisebb fontos részlet méretének ismeretében becsülhetjük meg. Próbáljunk olyan felbontást választani, hogy a legkisebb részletet legalább két pixel reprezentálja. A minimális pixelfelbontás becsléséhez alkalmazhatjuk a következő képletet:

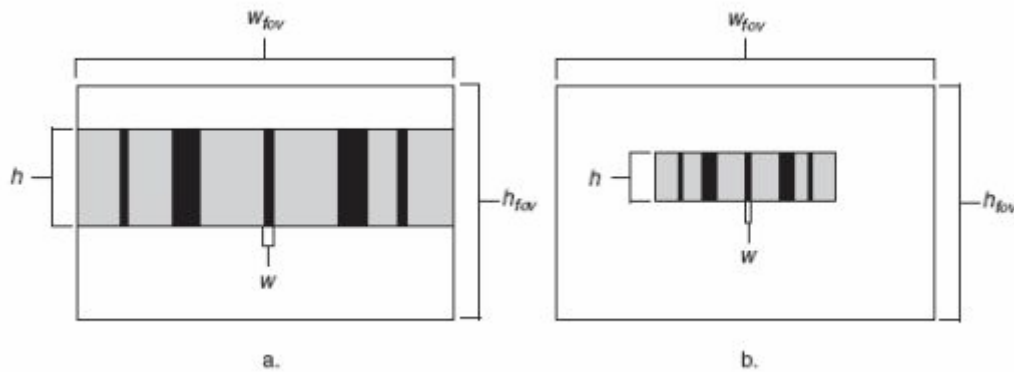
(az objektum leghosszabb tengelye/a legkisebb jellegzetesség)*2

Ha az objektum nem tölti be a teljes látómezőt, a kép mérete nagyobb lesz, mint a pixel felbontás.

A felbontás azt a legkisebb méretet jelenti a megfigyelt objektumon, ami még leképezhető a használt optikai rendszerrel. Rossz felbontású képen a részletek hiánya úgy jelentkezik, hogy a kép homályos. A képalkotó rendszer felbontását három faktor határozza meg, nevezetesen a látótér nagysága, a szenzor mérete, valamint a pixelek száma a szenzoron. E három paraméter ismeretében határozható meg a kamera lencserendszerének fókusz távolsága.

- Látómező

A látómező az a terület, amelyről a kamera képes képet készíteni. A következő ábra bemutatja mi az összefüggés a látómező mérete és a pixelfelbontás között:



Ha w a legkisebb méret a vízszintes x irányban és h a legkisebb méret a függőleges y irányban, akkor a minimális pixelfelbontás az x és y irányban a következő:

$$2 \frac{w_{fov}}{w}, \quad 2 \frac{h_{fov}}{h}$$

- Szenzor

A szenzor mérete fontos a látómező meghatározásához. A szenzor átlója adja meg a szenzor érzékeny tartományának méretét. A szenzor pixeleinek száma nagyobb, vagy egyenlőnek kell legyen a választott felbontással.

- A lencse

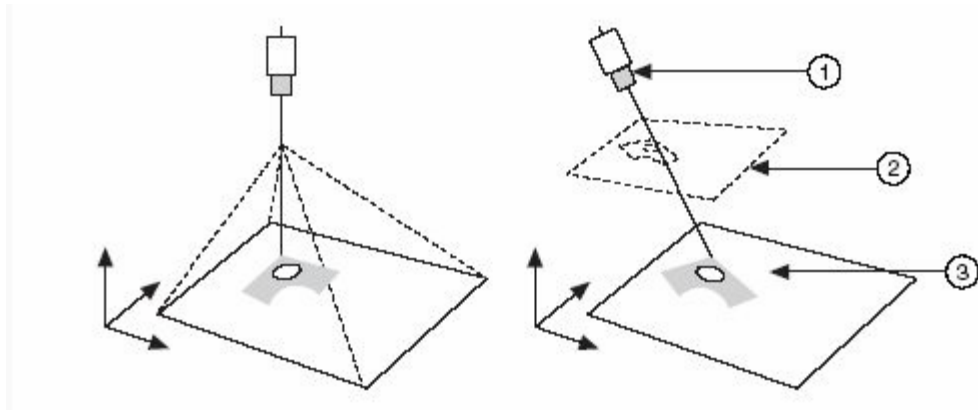
Az előbbi adatok ismeretében meghatározható milyen lencserendszer felel meg a mérőrendszer támasztotta igényeknek. A lencse fókusztávolsága és az előbb ismert paraméterek között a következő összefüggés áll fenn:

$$\text{Fókusz-távolság} = (\text{szenzorméret} * \text{munkatávolság}) / \text{látómező}$$

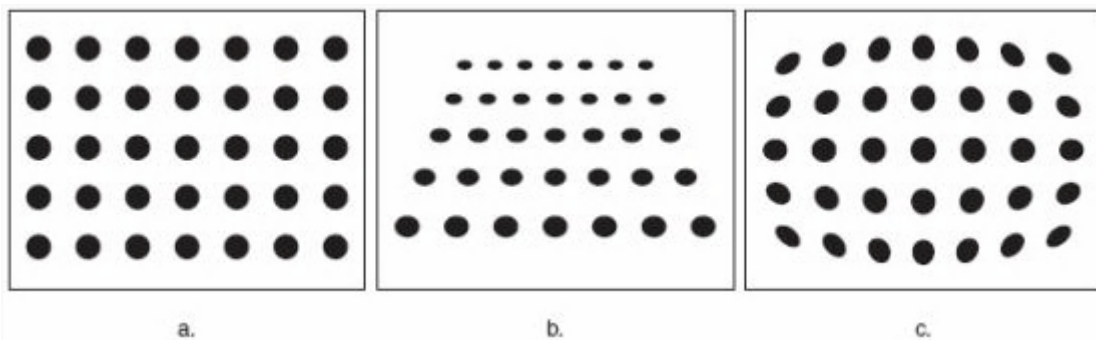
Ha a munkatávolság nem változtatható, a választható lencsék száma erősen korlátozott. Amennyiben azonban a munkatávolság változtatható, nemcsak nagyobb lesz a lencseválték, de az egyéb hibákkal is könnyebben birkózhatunk meg.

- Hibák és Korrigálásuk

Amennyiben a kamera tengelye nem merőleges a megfigyelt objektum síkjára, perspektivikus hibák lépnek fel (II.3., II.4.a.b ábra). Ezek a hibák könnyen korrigálhatók, amennyiben ismerjük a kamera tengelyének a vizsgálat síkjával bezárt szögét.



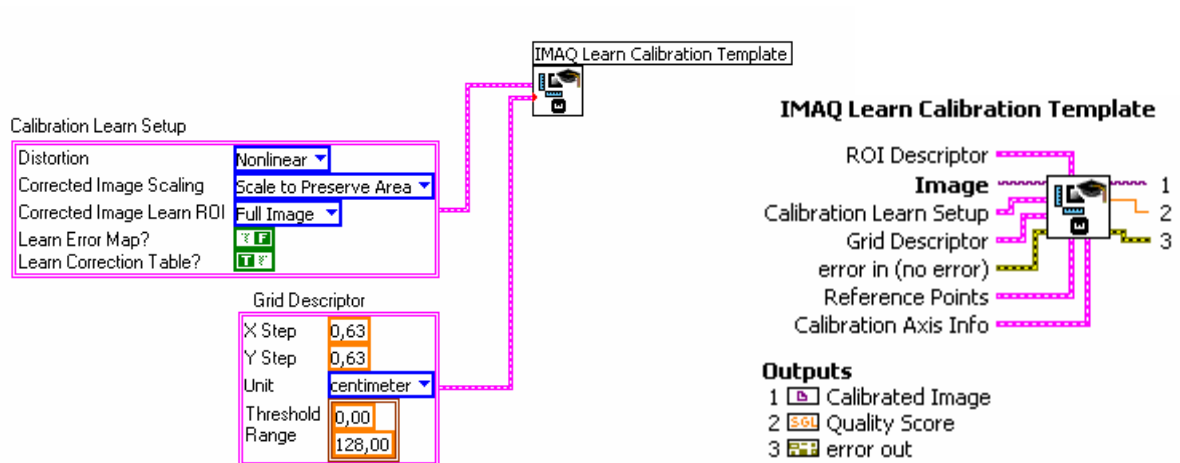
A lencse optikai jellegű hibái is képtorzulást okoznak, melyek a lencserendszer geometriai hibáiból adódnak. Ezek a hibák a képalkotó rendszer megfelelő elhelyezésével korrigálhatók.



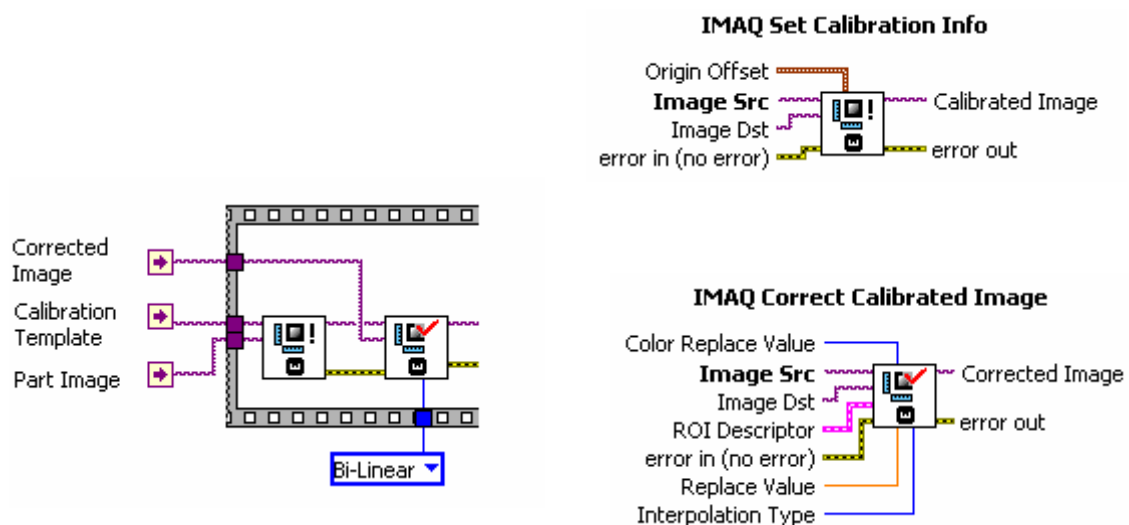
Egyéb megoldás is létezik ezen hibák korrigálására. A Labview-ban külön írhatunk rá programot, hogy a torzult képet korrigáljuk. Lineáris transzformációkat hajtva végre a képen, közel megfelelő minőségű képet kapunk.

Egy lehetséges algoritmus a kamera által készített kép kalibrálására:

1. Ugyan azon fókusz távolságon, kalibrációt használva készítünk egy képet az úgynevezett kalibrációs lapról. (CalibrationGrid).
2. Meghívjuk az IMAQ Learn Calibration Template függvényt.



3. A kalibrációs lapról és a minta képről leírást készít az IMAQ Set Calibration Info VI, majd a képet átadva az IMAQ Correct Calibrated Image VI –nak, egy javított képhez jutunk.



4. Program:

Elsődleges célom a fejlesztés során, egy olyan könnyen kezelhető program írása volt, ami lehetővé teszi a gyártósoron haladó késztermékek vizsgálatát egy megfelelően elhelyezett kamera segítségével. Programozás során a programnak két verzióját készítettem el.

1. Egy off – line program, ami nincs közvetlen kapcsolatban a kamerával.

Ezen modul fejlesztése során ismertem meg elégséges mélységben az IMAQ Vision azon beépített függvényeit aminek segítségével elő tudtam állítani a Real –Time time programot. Saját adatbázist használ, az input paraméterek fájlokból származnak.

2. Real – Time modul , ahogy a neve is tükrözi, azonnali eredményt ad, a kamera látómezejében áthaladó termékről. Lényegesen kevesebb beállítási lehetősége van, az Off – Line társához képest, hisz ez is a könnyebb felhasználhatóságot eredményezi.

A képeket mindkét esetben ugyan azzal a kamerával készítettem (Blaser asA 1600 – 14 GC), és így mind a két program tesztelése azonos eredményt adott.

A továbbiakban részletes útmutató olvasható a program megfelelő használatát illetően:

- Gépigény
- Futtató környezet
- Könyvtár szerkezet létrehozása
- Progamegységek
- Munkamenet

- Gépigény (ajánlott minimum konfiguráció):

1. 3200+ CPU 64bit
2. 2 GB Ram
3. 512 MB VGA

- Futtató Környezet:

1. +NI.Vision.v8.5.Acquisition.Software Labview 8.5 vagy frissebb verzió
2. +NI.Vision.Builder.AI.v3.0.1
3. +NI.Vision.v8.5.Development.Module Labview Vision Developement Modul
4. +Labview Drivers
5. +Labview 8.5 vagy frissebb verzió
6. Windows XP SP3

- Könyvtárszerkezet Létrehozása és Kötelező beállítások

Készítsünk egy fő könyvtárat, ami minden szükséges elemet tartalmazni fog, legyen a neve :END. Ezen a mappán belül négy alkönyvtárat hozunk létre:

1. „Koordinata_kepek”
2. „Keresendo_kepek”
3. „Vizsgalando_kepek”
4. „Tokeletes_kepek”

A „Vizsgalando_kepek”mappa csak az off –line modulhoz szükséges, viszont a többi mappa megléte elengedhetetlen.

1. Koordinata kepek:

Ebben a mappában tároljuk azokat a képeket, amik definiálni fogják a vizsgálandó területeket a lapkán.

Minden egyes vizsgálandó alkatrésztől készíteni kell egy templétet, amit lehet pl. a beépített Template Editorral készíteni, viszont ügyelni kell, hogy a helyes opciót kell kiválasztani a minta elkészítéséhez.

Pattern Matching mintákat hozunk létre. Használható az általam készített program a „2in1.vi„, amivel közvetlenül a megadott helyre menthetjük a mintáinkat.

Fontos kiemelnem, hogy ismétlődő alkatrészek esetén, (pl : két darab soros port van) mind a két eszköztől készíteni kell egy – egy mintát. Hosszadalmas munka, viszont elég ezeket egyszer elkészíteni és megőrizni.

2. Keresendo kepek:

Itt azoknak az elemeknek a képeit tároljuk, amik meglétére kíváncsiak vagyunk. Célszerű a Koordinata_kepek mappa tartalmát átmásolni ide, fontos, hogy az ismétlődő elemek képeit egyszer másoljuk át. (Pl.soros port közül az egyik hiányzik, viszont két minta is van ugyanerről a portról, mivel ezek tökéletesen megegyeznek, a program mind a két mintát illeszteni fogja az egyetlen soros port előfordulására, ami így téves információt szolgáltat.

Ezt a műveletet is elég egyszer elvégezni, a mintákat megtartani.

3. Vizsgalando kepek:

Ha az off – line modult szeretnénk tesztelni akkor kötelezően fel kell tölteni ezt a mappát a vizsgálandó lapok képeivel. Ezek a képeken történik a keresés, az eszközök meglétének vizsgálata.

Az on-line modul esetén, természetesen nincs szükségünk ezt a mappát feltölteni, hisz a vizsgálandó kész kártyák képet a kamera készíti, azok nem kerülnek mentésre, egyből a SubVi – bemenetére kerülnek.

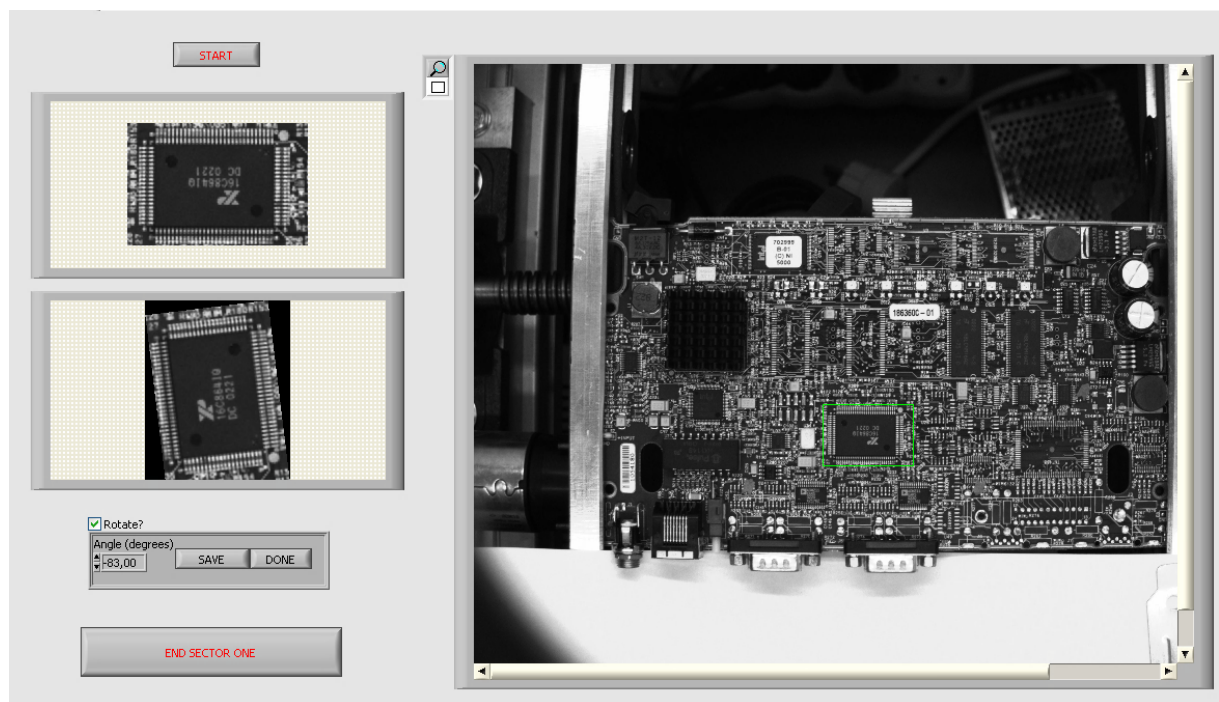
4. Tökéletes kepek:

Ide a referencia kép kerül, melyet egy olyan termékről készítetünk, amin minden alkatrész megtalálható. A program ehhez a képhez fogja hasonlítani az aktuálisan vizsgált termék képét, pontosabban azokon fellelhető alkatrészek meglétét. A hiba feltárásakor a tökéletes képhez viszonyítva mutatja meg a program a hiányzó vagy hibás alkatrészek helyét is.

- Progamegységek

- **Template_Creator(2in1.VI)**

Ez az alkalmazás saját fejlesztés, elsősorban Pattern Matching mintákat készíthetünk vele. A termékellenőrző programban használatos templéteket ezzel készítettem el.



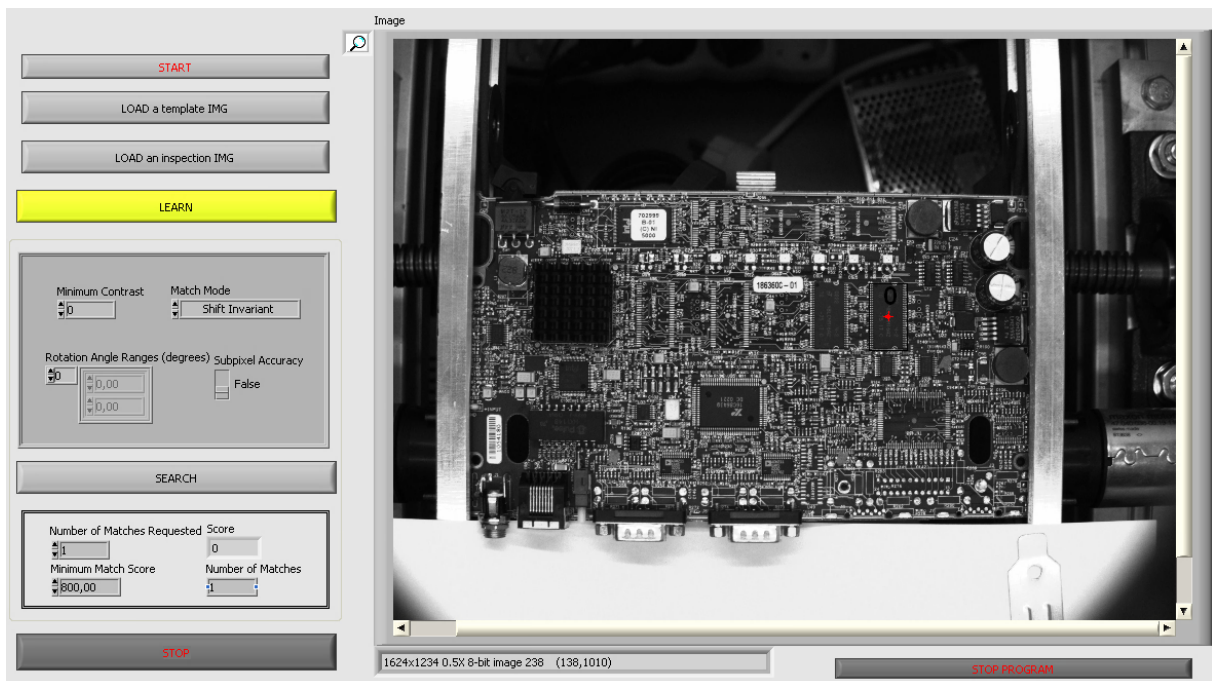
A START gomb lenyomásának hatására a program egy képfájl betöltésével kezd, majd a ROI függvény segítségével tetszőleges téglalap alakú területeket vághatunk ki. Ezeket a kivágott mintákat konfigurálhatjuk:

- elforgathatjuk
- BCG beállításokat végezhetünk el (Brightness, Contrast, Gamma)

Minden egyes szerkesztett képet elmenthetjük aktuális állapotában. Ezek lesznek a fő programban használt mintaképek.

A minták elkészítése végén, az „END SECTOR ONE” gombbal lezárhatjuk a minta készítő részt.

Ezeket, a mintákat le is kell ellenőrizni, hogy milyen illesztési értékkel szerepelnek. Akkor készítettünk jó templétet, ha az illesztés mértéke 950, vagy magasabb. Erről az értékről a második tabulátoron bizonyosodhatunk meg.



1. START gomb elindítja a modult, betöltjük az általunk készített mintát. (Load a template IMG)
2. A felületet, amin az előfordulást keressük (Load an inspection IMG).
3. A program elkészíti a teljes leíró attribútumát a mintának a LEARN hatására.
4. A táblázatból leolvasható az illesztés mértéke.

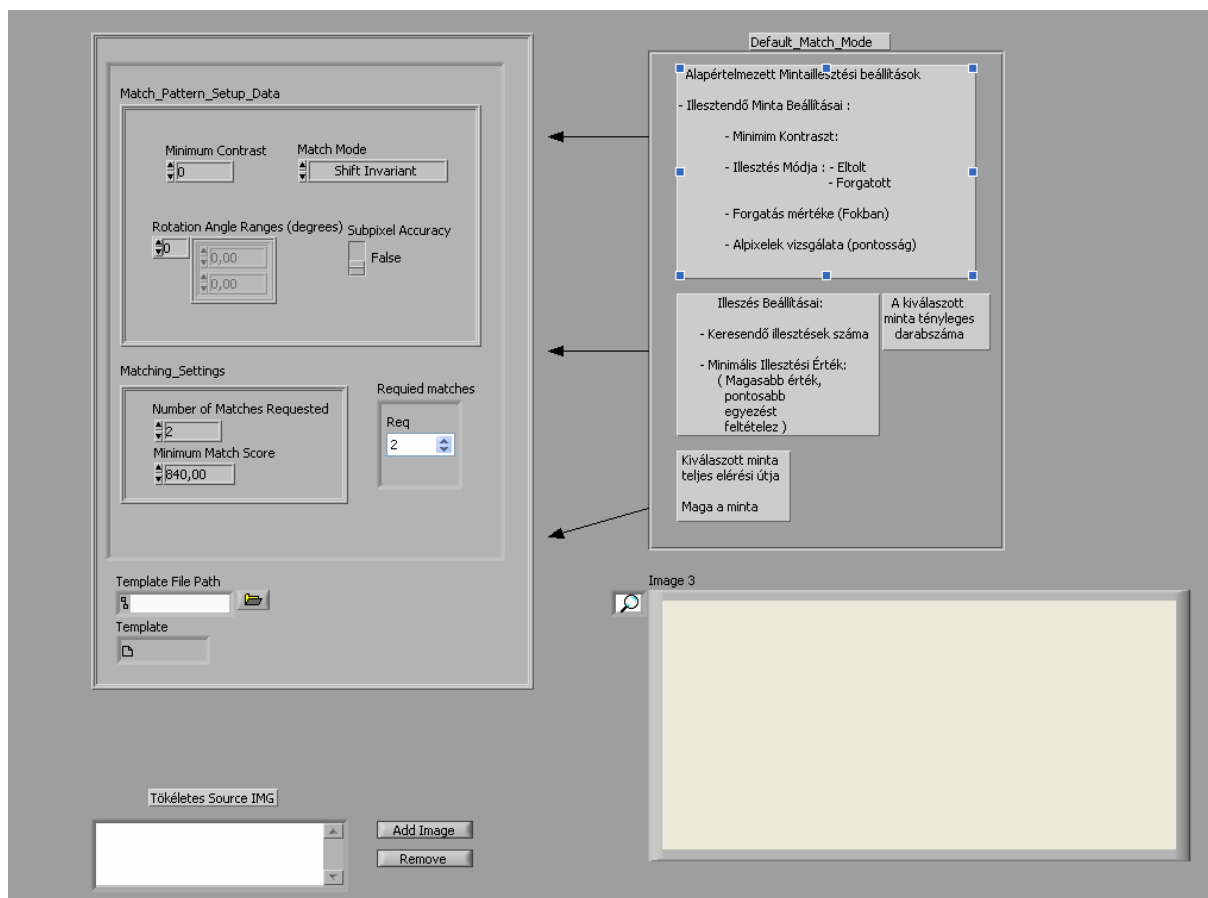
- **Config Deafault Match Mode**

(Első tabulátor)

Az itt beállított értékek az összes mintára vonatkoznak. Egyszerre tudjuk az összes illesztést és mintát konfigurálni.

Match Pattern Setup Data:

- Minimum kontraszt
- Illesztés módja:
 1. Eltolt
 2. Forgatott
- Forgatási illesztés során a forgatás mértéke szögben megadva
- Alpixelek vizsgálata



Matching Settings:

Az illesztésre magára vonatkozó paraméterek megadása. Alapértelmezett értéként.

1. Az elvárt találatok száma : 2

Minden egyes mintát illetően, mikor vizsgálunk, legalább ennyi előfordulást vegyen figyelembe a program. Tanácsos egy magas értékre állítani.

2. Minimális Találati Pontosság : 840

Illesztés során, minden megtalált alkat, ami illeszkedik a mintára egy illesztési értéket kap. Tökéletes illesztés során ez az érték maximálisan 1000 lesz. Ez akkor érhető el, ha az általunk készített minta jó minőségű, és a kamera is helyesen van fókuszálva a vizsgált lapra. Itt szigorítható maga az illesztés, és kiszűrhetőek vele az ugyan hasonló de mégsem egybevágó előfordulások felkutatása.

3. Required matches : A tényleges előfordulás, ez a Search_Selected_Templates tabulátor alatt található táblázat „Találatok” értékét állítja be a kívánt értékre.

Használata: Config Deafault Match Mode

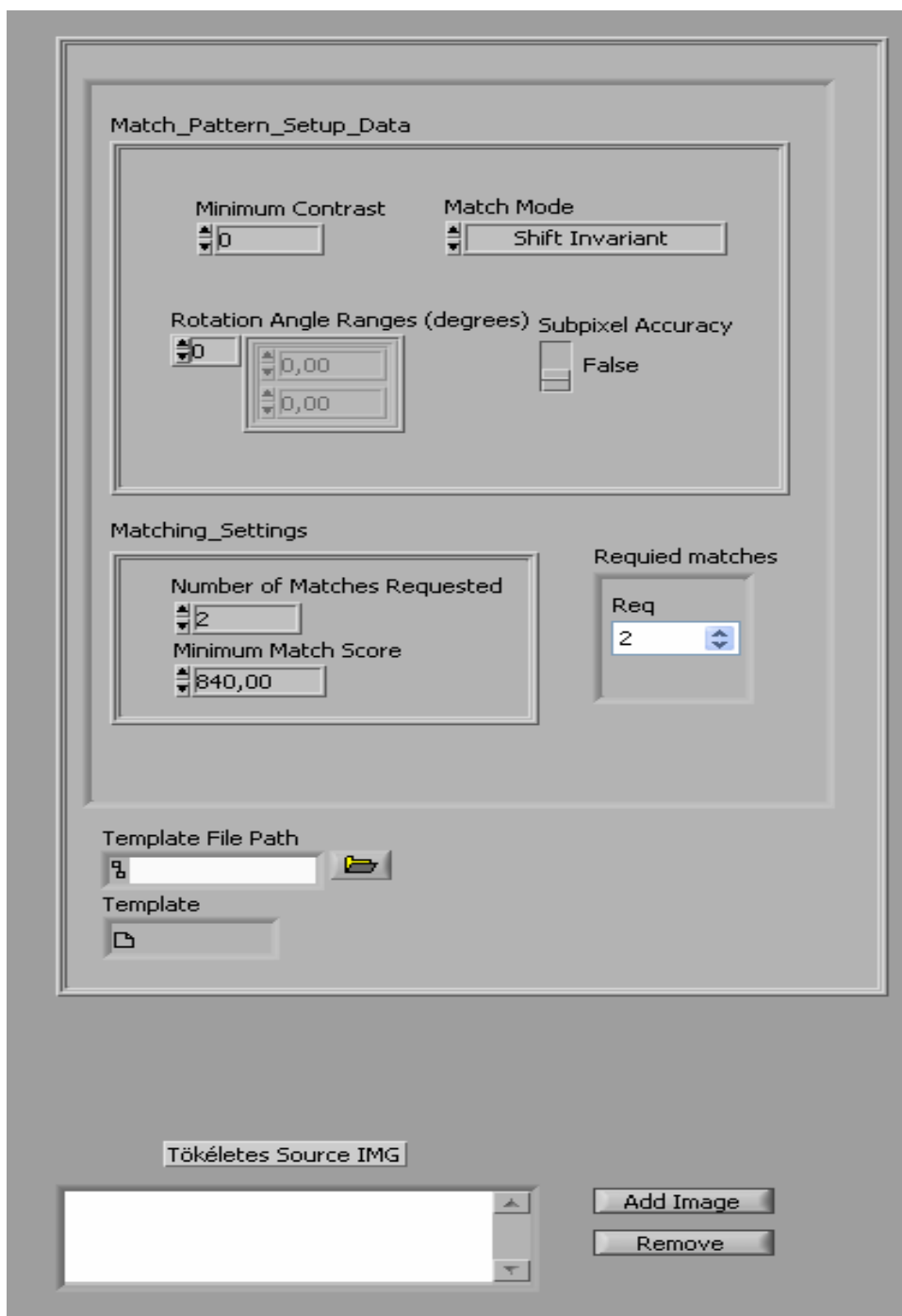
A program a működését tekintve, két eltérő, de mégis szorosan összekapcsolódó feladatot láthat el. Attól függően, hogy melyiket akarjuk elvégeztetni a programmal, úgy kell konfigurálni ezen alapértelmezett beállításokat.

Az egyik lehetséges feladata, az általunk keresett elemek felkutatása a nyákon, és azok azonosítása. Táblázatban az értékek megjelennek és megbizonyosodhatunk arról, hogy a keresett elem,elemek tényleg szerepelnek a lapon.

A másik pedig maga a hiányzó vagy sérült elemek megkeresése és helyük azonosítása, a tökéletes laphoz viszonyítva. Ebben az esetben, számunkra az az elsődleges cél, hogy a keresett alkatrészek szerepelnek e, vagy sem, akkor a Number of Matches Requested értéket úgy kell beállítani, hogy értéke, a NYÁK-on fellelhető, egy mintára eső vizsgálat során, a legnagyobb előfordulás legyen.

Tökéletes Source IMG:

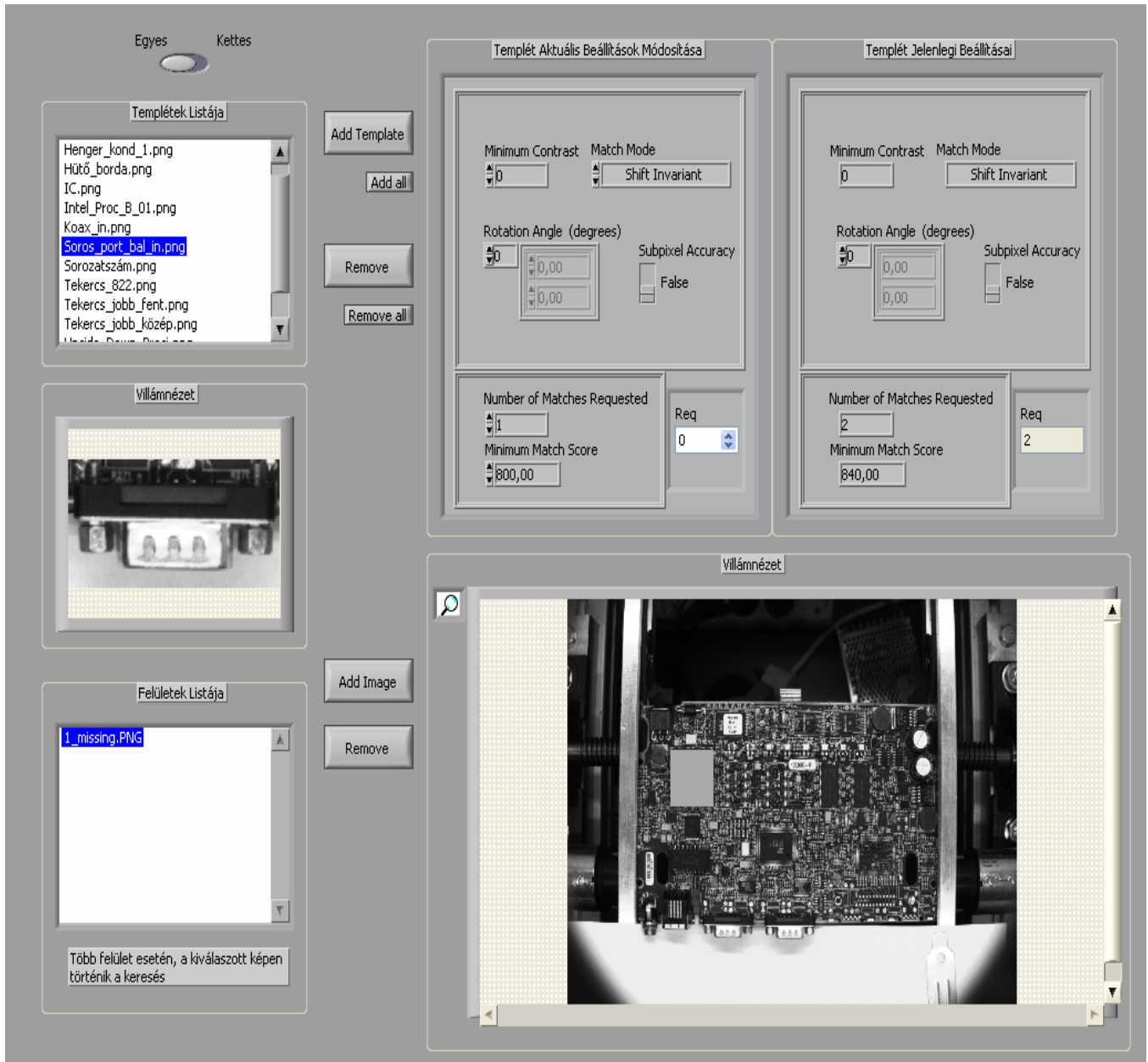
Itt adható meg a hiányzó alkatrészek felkutatásához szükséges kép, amihez hasonlítjuk az aktuálisan vizsgált terméket. A hiányzó eszközök ezen a képen jelennek majd meg piros négyzettel jelölve.



A fenti képen, a program első tabulátorát lehet látni. Itt adhatóak meg az alapértelmezett mintaillesztési beállítások.

- **Select Templates to Investigate**

(Második tabulátor)



A képen látható a program második tabulátorának tartalma

Balra, fent lévő Templétek Listájában találhatóak azon elemek, amikre a keresést el akarjuk végezni.

- *Add Template*: Egyesével adhatjuk meg a vizsgálandó objektumokat a „Keresendo_kepek” mappából.
- *Add all*: Betölti a „Keresendo_kepek” mappa tartalmát.
- *Remove / Remove all* : vezérlőkkel eltávolíthatjuk az elemeket a listából.
- A *Templétek Listája* listboxban az aktuálisan kiválasztott mintának az alapértelmezett illesztési beállításait definiálhatjuk felül, ezzel lehet az egy mintára vonatkozó keresési feltételeket egyedivé tenni. Ez lényegében az első tabulátoron elvégzett alapértelmezett mintaillesztési állapotokat változtatjuk meg.

A bal oldali vezérlőn adhatjuk meg az illesztési megszorításokat, összehasonlítva a mellette lévő indikátorral, ami az eredeti állapotot mutatja.

A *Felületek Listája*, bal lent, az off – line illesztésnél van szerepe. Itt adhatjuk meg, hogy melyik képen végezzük a vizsgálatot.

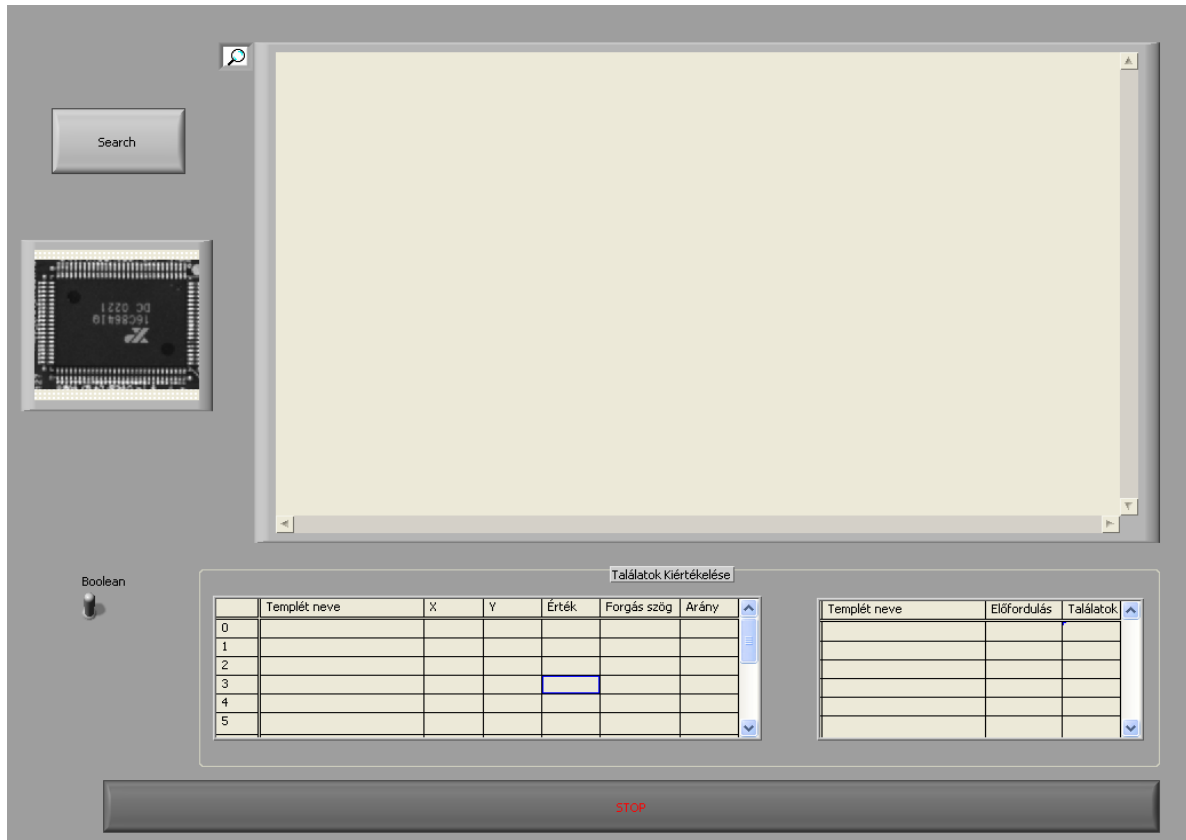
Add Image :gombbal adhatóak a képek és az kerül vizsgálatra amit kijelölünk egérrel.

Remove : Eltávolítjuk a vizsgálandó képet a listából.

Villámnézet ablak segítségével tudjuk beazonosítani, hogy melyik képet vizsgáljuk.

A Real – time vizsgálat során, ez a funkció nem él, hisz az aktuálisan vizsgálandó képet közvetlenül a kamera szolgáltatja. Ebben az esetben nincs szükség a kép szerializációjára és ezáltal nem is kell megnyitni.

- **Search Selected Templates**
(Harmadik tabulátor)



A kép a keresés előtti állapotot mutatja. A bal oldali kijelző a keresendő mintát mutatja.

A *Search* gomb elvégzi a keresést és a címkézést is egyszerre.

A *Kapcsoló* fölkapcsolásával nemcsak grafikus, hanem numerikus információt is kapunk az előfordulásokról.

Találatok kiértékelése:

Bal oldali táblázatból megtudhatjuk, hogy a minta milyen névvel szerepel a fájlban, mi az előfordulásának a pozíciója, az illesztés értéke (mennyire egyezik meg a minta az képen levő előfordulással), ha elforgatott előfordulást vizsgálunk, akkor a minta és az előfordulás egymáshoz képest lévő szöge, nagyítási aránya, ha a minta méreteiben kisebb vagy nagyobb

mint az előfordulás. A táblázat sorindexe megegyezik a kijelzőre felrajzolt egyezés sorszámával, a könnyebb azonosítás végett.

Jobb oldali összegző táblázatból leolvasható, melyik minta hányszor szerepel a képen. Az előfordulás gyakoriságát nekünk kell beállítanunk az első és a második tabulátoron (ott már egyedileg megszabva).

Összevetve a találatok adatait az előfordulással, észrevehető, ha valamelyik minta nincs meg az előfordulása a képen.

Egy keresési eredmény:

The screenshot shows a software interface for image search. The main window displays a grayscale image of a circuit board with 13 green bounding boxes and red crosses, each labeled with a number from 0 to 12. To the left is a 'Search' button and a thumbnail of a component. Below the image is a 'Boolean' toggle. At the bottom, there are two tables: 'Találatok Kiértékelése' and a summary table.

Templét neve	X	Y	Érték	Forgás szög	Arány	
0	Függőleges_Proc_jobb.png	1063,00	540,00	1000,00	0,00	1,00
1	Függőleges_Proc_jobb.png	948,92	539,17	844,88	0,00	1,00
2	Henger_kond_1.png	1265,00	407,00	1000,00	0,00	1,00
3	Henger_kond_1.png	1275,91	486,00	950,60	0,00	1,00
4	Hűtő_borda.png	452,50	513,50	1000,00	0,00	1,00
5	IC.png	439,50	754,00	1000,00	0,00	1,00

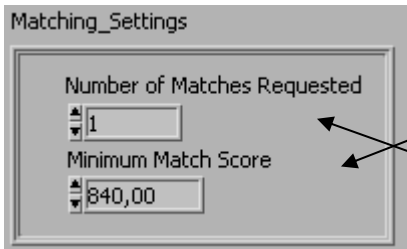
Templét neve	Előfordulás	Találatok
Függőleges_Proc_jobb.png	2	2
Henger_kond_1.png	2	2
Hűtő_borda.png	1	1
IC.png	1	1
Intel_Proc_B_01.png	1	1
Koax_in.png	1	1

A jól elkészített minta 1000 értékű egyezést mutat. Megfigyelhető még, hogy a Henger_kond_1.png értéke első esetben 1000, majd 950,60.

Ez abból adódik, hogy a „Keresendo_kepek” mappában egyszer szerepeltettük a Henger_kond_1.png fájlt. Az elkészített minta a 2. sorszámú találatról készült, mert az egyezés ott a magasabb. A 3. sorszámú egyezés alatti érték azért alacsonyabb, mert az a minta ugyan illeszkedik arra a képre is, de nem olyan pontossággal. Ez egy előfordulása a mintának.

További beállítások hatása az eredményre:

Mikor először használjuk a programot, akkor tanácsos az összes minta keresése egy tökéletes képen. Majd a táblázat *Érték* oszlop értékeinek a minimumát venni. Ez az érték legyen a következő futás, első tabulátorán elhelyezett *Minimum Match Score* értéke.



Rutinosabb felhasználók a *Number of Matches Requested* értékét is a találatokhoz igazíthatja.

A meg nem talált illesztést a táblázat üres értékkel jeleníti meg.

Templét neve	Előfordulás	Találatok
Tekercs_822.png	1	1
Tekercs_jobb_fent.png	1	1
Tekercs_jobb_közép.png	1	1
Upside_Down_Proci.png	1	1
USB_in.png	1	

Ezen három tabulátor alkalmazásával elérhető, hogy elemek előfordulását vizsgálhatjuk le, és az értékekből következtetéseket vonhassunk le. Mielőtt áttérünk a hibás, hiányzó alkatrészek felkutatására, érdemes ezt a műveletet egyszer végrehajtanunk. Megfigyelni, hogy minden általunk kívánatos eszközt figyelembe vesz e a program.

A táblázatbeli értékek és a vizsgálat alatt álló elemek egyértelműen azt mutatják, hogy elegendő mintát készítettünk, minden olyan területet figyelembe vettünk, ami az elvárás volt. Az egyezések illesztési értékei elégségesen magasak voltak ahhoz, hogy ezeket a mintákat későbbi felhasználásra megtarthassuk.

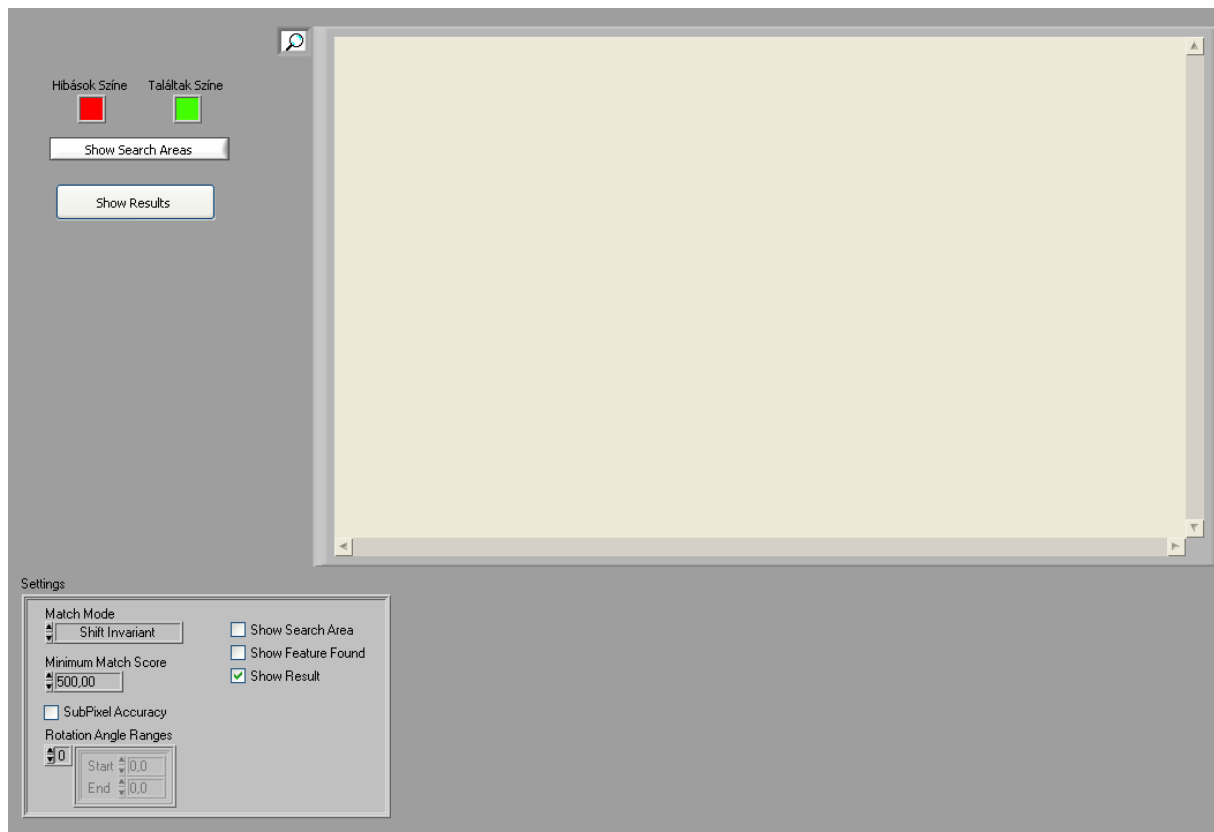
Tegyük fel, hogy valamelyik minta nem elégségesen illeszkedik a képre. Ha a vizsgálandó kép nem lenne ilyen jó minőségű, akár az egyezést sem ismerné fel a program. Megfordulhat a fejünkben ezen eset kezelése, miszerint állítsuk a minimális illesztési értéket kellően alacsonyra, hogy kiküszöböljük a hibát.

Ez olyan eshetőséghez vezetne, miszerint olyan elemekről állítaná a program az egyezést, ami valójában nem is egybevágó a mintával, hanem csak szimplán hasonló. Köztudott, hogy ilyen eszközök vizsgálatakor magas precizitás van megkövetelve a gyártótól, és ezáltal az ellenőrző részlegtől ,tehát a programról.

Ezért is nem tanácsos az illesztési minimum értéket alacsonyra állítani. Ideális illesztésről beszélünk, ha az illesztési érték 1000 – 900 közötti. 900-800– as intervallum még elégséges ahhoz, hogy az elforgatott, eltolt minták, és az „egy minta – többszörös egyezés” esetét megfelelően kezelje.

- **Show Results**

(Negyedik tabulátor)



Eredmények megjelenítése, melyben a hiányzó és meglévő elemek hollétéről kapunk információt. A meglévő és hiányzó elemekről színek alapján informálódunk, amit a felhasználó kedve szerint választhat meg. Összevetésre kerülnek az aktuális képen talált és a tökéletes lapon lévő alkatrészek.

Azért nem magának a két képnek az egészét hasonlítja össze a program, mert akkor nem lehetne kimondottan bizonyos elemek meglétére koncentrálni.

Ha kimondottan csak egy elemre (pl: hűtőbordára) vagyunk kíváncsiak, hogy az szerepel-e az aktuális elemen, viszont nem vagyunk tekintettel a más elemre (pl.: kondenzátorok), akkor az előbb említett illesztés elmélete nem felelne meg az elvárásnak. Ezért döntöttem úgy, hogy a programot olyan módon írom meg, ami figyelembe veszi a felhasználó által kiválasztott elemeket és nem az egész képet magát.

Áttérve erre a panelre, az előző illesztés eredményei láthatóak, egy utolsó szemrevételezés, egyeztetés miatt.

Kapcsolók:

Show Search Area: Ezzel lehet leellenőrizni, hogy az illesztés során, mely területek lesznek vizsgálat alá vetve. Nem tévesztendő össze ezen ellenőrzés az előző Search Select Templates – részben lévő ellenőrzéssel.

A lényeges különbség a két ellenőrzés között, hogy míg a kereső résznél lévő ellenőrzés során arról kapunk információt, hogy az elem megtalálható az aktuális képen, addig ezen ellenőrzés pedig a „Koordinata_kepek” mappában lévő elemek elégséges mennyiségéről nyújtanak információt.

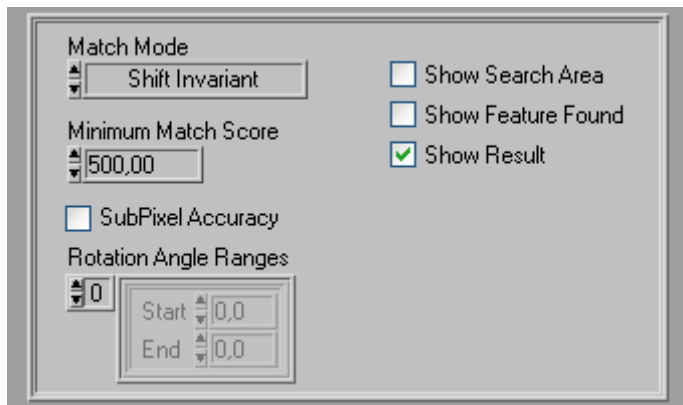
Tehát, ha találunk olyan területet, ami nincs vizsgálat alá véve, akkor készítünk egy Pattern Matching mintát azon területről, ahol azt végre akarjuk hajtani. Ezáltal egy új koordináta tengely jelenik meg a képen.

Show Results: 2- 3 másodperc várakozás után, a kijelzőn megjelenik a tökéletes kép, amit a „Tökéletes_Kepek” mappából olvastunk be az első panelen. Azon elemek, amik szerepelnek az aktuálisan vizsgált képen azokat egy zöld négyzet jelöli. Ez azt jelenti, hogy az elem tényleg megtalálható a NYÁK-on. Viszont, ha piros négyzet jelenik meg, akkor a ott hiányzó elem van. Az összehasonlítás alapja, a tökéletes képen találhatóalkatrészek és az aktuálisan vizsgált képen levő alkatrészek összevetése.

Ezáltal, ha egy elem nincs meg, akkor annak a helyét a tökéletes képen lévő pozíciójával veti össze.

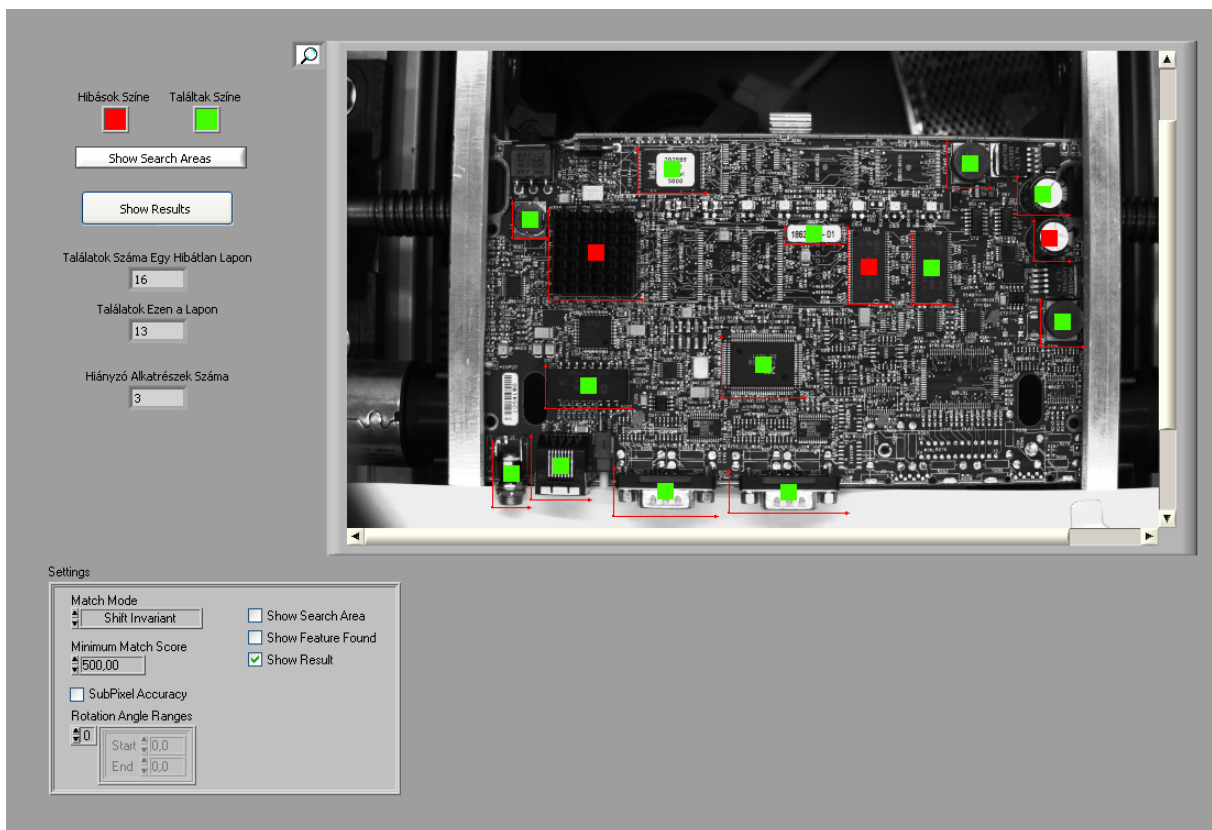
A kijelzőn a tökéletes kép látható, és a piros négyzet pedig a vizsgált képen lévő hiányt mutatja a tökéletes képhez viszonyítva.

Settings:



Mikor a vizsgálandó képünk elforgatott, akkor kell az itt lévő beállításokon változtatni.

Az itt található Minimum Match Score, értéke lényegében nem mérvadó. Ez az érték felel a „Koordinata_kepek” mappában lévő minták illesztése a tökéletes képre funkcióért.



A programnak egy lehetséges kimenete

A képen jól látható, hogy olyan lapot vizsgál a program, amin 3 darab hiányzó alkatrész van. A Show Search Areas kapcsolót használva kirajzolódó koordináta tengelyek hollétére tudunk következtetni arra, hogy melyek azok a területek ahol a vizsgálatot végrehajtjuk.

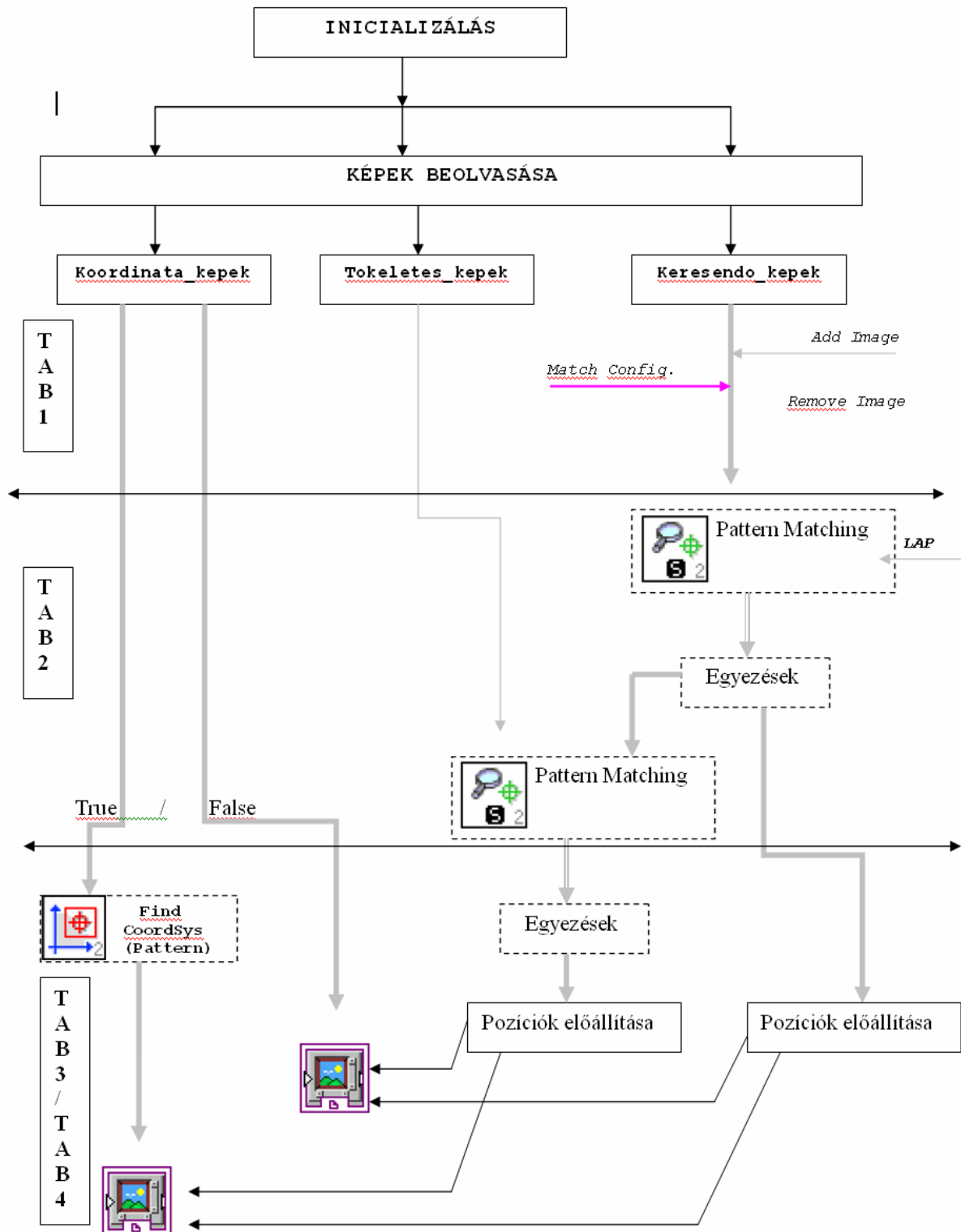
- Munkamenet:
 - 1.) Kiválasztjuk a keresendő elemeket, feltöltjük a Templétek Listát (Add Templates , Add All). (tab_2)
 - 2.) Off – Line esetén : a vizsgálandó lapok kiválasztása
Real – Time esetén: ez automatikus (tab_2)
 - 3.) Kiválasztjuk a tökéletes képet (tab_1)
 - 4.) Előfordulások keresése (tab_3)
 - 5.) Hiányzó alkatrészek megjelenítése (tab_4)

5. Algoritmus:

1. Inicializálás: Azon adatszerkezetek létrehozása, amik minden szükséges információt tartalmaznak a mintaillesztéshez, táblázat feltöltéséhez, koordináták, elérési utak tárolása és a program előző futási eredményeinek törlése.
2. Képek mappáinak ellenőrzése, minták, felületek képei beolvasása egyenként, egyszerre. Használni nem kívánt képek törlése a listából, egyéni összeválogatás. Alapértelmezett mintaillesztési beállítások megadása, majd ezek felüldefiniálása egyenként. Táblázathoz szükséges adatok felvitele. Off-line mód esetén, a vizsgálandó lapka képeinek betöltése. Tökéletes kép kiválasztása, beolvasása.
3. A vizsgálandó lapka felületére elvégezzük az illesztést, a minták segítségével. A talált illesztéseket letároljuk és megjelenítjük a kijelzőn. Sorszámmal látjuk el a mintákat, a megtalálási sorrendet figyelembe véve. A táblázatot feltöltjük adatokkal.
4. Az illesztés során előálló minták mindegyikét illesztjük a tökéletes képre, az egyezést szintén letároljuk.
5. A „Koordinata_kepek” mappa tartalmára meghívjuk a Find CoordSys (Pattern) 2 függvényt, ezek definiálják a keresendő felületeket.
6. Kirajzoljuk az összes illesztés eredményét, amit a tökéletes képen végeztünk el, és az egyezéseket megjelenítjük egy adott színnel (piros).
7. Azon egyezések eredményét, amik az aktuális lap vizsgálata során álltak elő, a koordináták alapján ezeket az illesztéseket is megjelenítjük a tökéletes képen, egy másik színnel.(zöld)

8. Azon felületek, amik nem lettek átszínezve zöldre, tehát pirosak maradtak, azokon a pozíciókon nincs előfordulása a mintának, nem található ott az a bizonyos alkatrész, a vizsgált lap Selejt.

Algoritmus ábra



6. Összefoglalás:

Témaválasztásom során, az elsődleges szempont az volt, hogy olyan témakörben kutathassak, ami Labview-val és legfőképp az IMAQVision Modullal kapcsolatos.

Jelen álláspont szerint, nagyobb szükség van egy kisebb költségvetésű optikai vizsgáló rendszerre, mint egy nagyobb anyagi háttérrel igénylő társára. A Labview segítségével megvalósítható mindez. A projekt célja egy olyan termékvizsgálatra alkalmas vizsgálóállomás létrehozása ami alacsony költségvetésű ugyan, de magas minőséget és pontosságot ad. Véleményem szerint, az általam készített program alkalmas az átmenő-furatos áramkörü elemek vizsgálatára. Tervezem ezen program tovább fejlesztését. Elsődleges cél a futási idő redukálása és magasabb pontosság elérése. A következő verziójában a program legalább két kamerát fog párhuzamosan kezelni. Terveim közt van egy harmadik, infra-képet adó kamera beiktatása az ellenőrzésbe is. A kamerák összehangolása párhuzamos programozást igénylenek, amit a Labview-ban könnyen meg lehet valósítani. Kibővíteni egy Classification modullal a programot, amivel az osztályozás valósítható meg.

Szeretném elkészíteni a következő verzióját a programnak, oly módon, hogy emberi beavatkozást minimálisra csökkentse. Az osztályozást (Classification) segítő függvények segítségével megvalósítható az áramkör beazonosítása és elemeinek mintái explicit módon való betöltése az adatbázisba. Ennek segítségével, nincs szükség minden új típusú lap vizsgálatkor a megfelelő adatbázis betöltése.

Véleményem szerint, egy szerver – kliens alkalmazás is építő ötlet lehetne ilyen vizsgálatkor. Egy központi szerverbe töltené fel a vizsgált NYÁK paramétereit, és a vizsgálat eredményét, ami, felhasználható statisztikák készítésre, hiba–arány meghatározására, esetleg egy rámutathat a gyártási technológia esetleges változtatásának szükségességére.

A programírás során azért döntöttem angol nyelvű szavak, feliratok használata mellett, mert a program egy olyan projekt része, mely végül a National Instruments debreceni gyáregységébe kerül, ahol ez az alapvető elvárás.

7.Irodalomjegyzék:

[1] Labview referencia

<http://zone.ni.com/reference/en-XX/help/371361A-01/>

[2] IMAQ Vision for Labview Users Manual; NI kiadvány (2004)

[3] S.Ertük: Digital Image Processing; NI kiadvány (2003)

[4] NI Vision Concepts Manual; NI kiadvány (2005)

[5] Cserhádi Csaba: Műszaki képfeldolgozás Labview-val elektronikus jegyzet

<http://roller.ttk.unideb.hu/moodle/mod/scorm/player.php?a=30¤torg=eXeDIP46107d3c1bb275ade62&scoid=231>

[6] Székely Vladimír: Képfeldolgozás (Műegyetemi Kiadó) (2007)

[7] Álló Géza, Hegedűs Gy. Csaba, Kelemen Dezső, Szabó József: A digitális képfeldolgozás alapproblémái, Akadémiai Kiadó Budapest 1989

[8] Gácsi Zoltán, Sárközi Gábor, Réti Tamás, Kovács Jenő, Csepeli Zsolt, Mertinger Valéria: Sztereológia és képelemzés Miskolci Műszaki Egyetem, egyetemi tankönyv 2001

8.Köszönetnyilvánítás:

Szeretném megköszönni Dr. Cserhádi Csaba Tanár Úrnak és Dr. Szabó István Tanár Úrnak, hogy elvállalta szakdolgozatom témavezetői szerepét és építő jellegű tanácsaival, hasznos észrevételeivel és szakmai szaktudásukkal koordinálták diplomamunkám elkészítését.