

LiSEC ITALIA
- Automation | Software -

GUS

Controllo qualità del vetro

Il controllo della qualità nel mondo del vetro

Premesse

Il controllo della qualità è un processo sempre più importante ed essenziale nel mondo dell'industria in genere, questo processo fondamentale per il prodotto non deve però incidere sulla produttività dell'azienda o almeno deve essere limitato.

I processi che controllano la qualità di un prodotto, non devono e non possono essere di tipo soggettivo, *a discrezione dell'operatore che esegue la funzione*, ma devono seguire degli standard per una uniformità del prodotto finito a tutte le regole che, definite da normative nazionali ed internazionali, rendono il mercato il più standard possibile.

Nel mondo del vetro piano è sempre più diffuso il controllo ottico della forma/difettosità dei prodotti in modo automatico, la complessità di questa operazione è dovuta al fatto che gli stati di rilevazione e analisi devono avvenire in una tempistica tale che non abbassi la normale produttività dell'azienda.

Essendo oggi la tolleranza sempre più selettiva e la tempistica sempre più stretta, le aziende usano controlli informatici che “decidono” o aiutano l'operatore nella scelta.

Durante il processo produttivo il vetro viene movimentato e può quindi essere accidentalmente graffiato o sporcato, senza escludere la possibile difettosità della materia prima che può contenere impurità.

A tale proposito, la soluzione adottata oggi nel settore del vetro, è la “scansione” di un pannello di vetro poco prima dell'ultima fase di lavorazione, detta assemblaggio e subito dopo la fase di lavaggio.

La scansione viene effettuata attraverso un vero e proprio scanner (hardware) che acquisisce l'immagine del vetro.

L'immagine viene analizzata da un software e sottoposta a determinati controlli, al fine di essere catalogata come “conforme” o “non conforme”.

Oggetto del capitolato

Il progetto richiede la costruzione di una applicazione / utilità/ sistema software, applicabile specificamente al settore del vetro, che dovrà avere le seguenti caratteristiche:

- facilità d'uso (interfaccia user friendly);
- gestione di "recipes" (ricette) per il controllo delle non conformità (varie tipologie);
- Alerts e Reporting (errori e reportistica);
- Fruibilità via web (modulo esterno per);

Efficacia, e facilità d'uso, sono le caratteristiche principali che vengono richieste al prodotto.

Efficacia:

Il prodotto deve poter gestire il 100% delle casistiche, prevedendo qualsiasi tipo di non conformità.

Facilità d'uso:

L'utilizzatore finale deve essere in grado di gestire e configurare in modo facile e veloce l'intera interfaccia.

Lo scopo finale, dunque, è un prodotto flessibile ai cambiamenti e facile da utilizzare.

Scopo del software

Gestione delle “recipes”

Il sistema informatico deve essere in grado di analizzare l'intera immagine e classificare i difetti. Le “difettosità” sono tipicamente di: Dimensione, Forma, Intensità

Dimensione

Un difetto può essere rilevante o non rilevante in funzione della sua dimensione, dovremo quindi poter parametrizzare la nostra analisi in funzione della dimensione del difetto. E' ragionevole pensare che un difetto con una dimensione inferiore a 1mm può non essere rilevante, al contrario di un difetto che ha una dimensione superiore alla scelta della qualità di produzione.

Forma

Un difetto deve poter essere catalogato e riconosciuto a seconda della sua forma in modo da aiutare ad individuare la posizione nel prodotto in analisi, uno striscio deve essere visualizzato dal sistema in modo diverso da un punto che a sua volta è diverso da un difetto che ha una dimensione importante (errore ad area). Essendo il lavaggio del vetro il processo precedente alla scansione, dovrebbe essere altresì possibile identificare un eventuale goccia d'acqua dovuta ad una non corretta asciugatura, in modo da poter indicare a chi controlla che tale difetto era presente al momento della scansione ma subito dopo è “sparito” essendosi il vetro asciugato nel tempo trascorso dalla scansione al controllo (pochi secondi ma sufficienti alla sparizione di tale falso difetto)

Intensità

Un altro parametro fondamentale all'analisi è l'intensità di tale difetto, cioè quanto è visibile il difetto nel nostro prodotto. Possiamo immaginare quanto può essere fastidioso un punto nero se pur di dimensioni limitate in un vetro trasparente.

IMAGE1



IMAGE1: esempio di rappresentazione grafica delle difformità trovate su un pannello di vetro

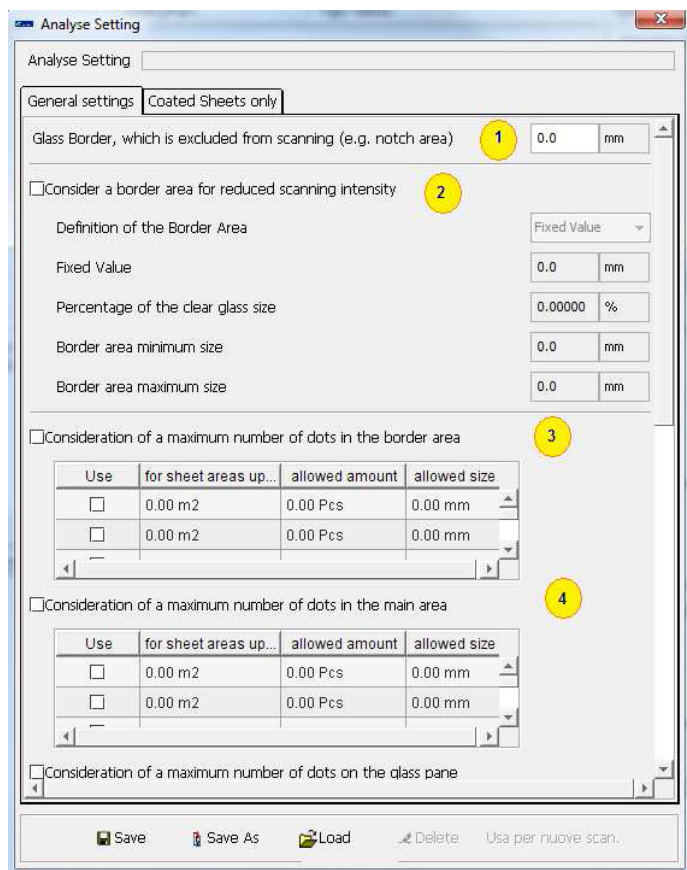
Ricette

Durante la fase produttiva l'analisi del prodotto può essere più o meno sensibile. Come in ogni processo produttivo possiamo avere necessità di produrre un prodotto di alta qualità e quindi più selezionato o decidere di produrre un prodotto standard (accettando una difettosità superiore). Tale ricetta può variare anche per tipologia di prodotto (per un vetro economico o meno economico posso decidere di variare la mia sensibilità) o per cliente finale (più o meno esigente).

Definizione ricette

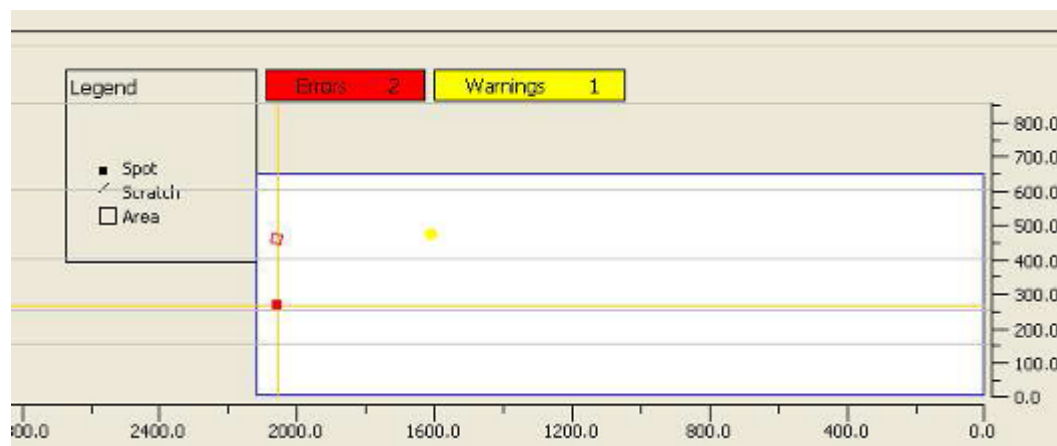
Le ricette devono poter permettere la configurazione secondo le norme in vigore o in funzione di alcuni criteri, di seguito alcuni esempi :

- a) Se ho una serie di errori "irrilevanti" perché di dimensione piccola o con un'intensità irrilevante, è ragionevole pensare che il sistema mostri questi errori anche se irrilevanti, nel caso in cui tali errori siano raggruppati in un'area limitata che si possa definire.
- b) Per normativa e per produzione possiamo dividere l'area da analizzare in 3 macro aree, bordo esterno, bordo interno, centro. Tali aree possono avere ricette diverse poiché un difetto sul bordo esterno (< 15 mm) non ha alcuna rilevanza, mentre lo stesso difetto potrebbe essere rilevate in una posizione più interna o non accettabile al centro.
- c) La quantità di errori potrebbe essere trascurabile o rilevante in funzione della dimensione del vetro (area in mq)



Alerts e Reporting

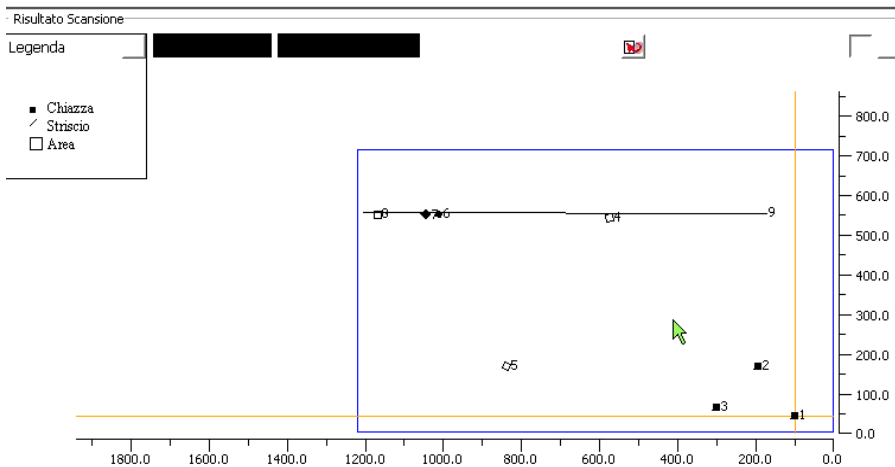
E' utile inoltre definire due livelli di segnalazione (errore o difetto); un errore è un'imperfezione con una intensità rilevata elevata, mentre un difetto avrà un'intensità inferiore.



Memorizzazione immagini

Ai fini di ricerca e report è necessario memorizzare l'immagine, essendo però l'immagine "grezza" di analisi molto grande in termini di spazio è necessario ideare un sistema che salvi non l'intera immagine, ma solo le informazioni basilari al fine di poterla ricreare.

Ad esempio salvando a database dimensione, quantità e posizione di errori possiamo ricreare l'immagine salvando solo gli errori rilevati (piccola porzione dell'immagine)



Statistiche

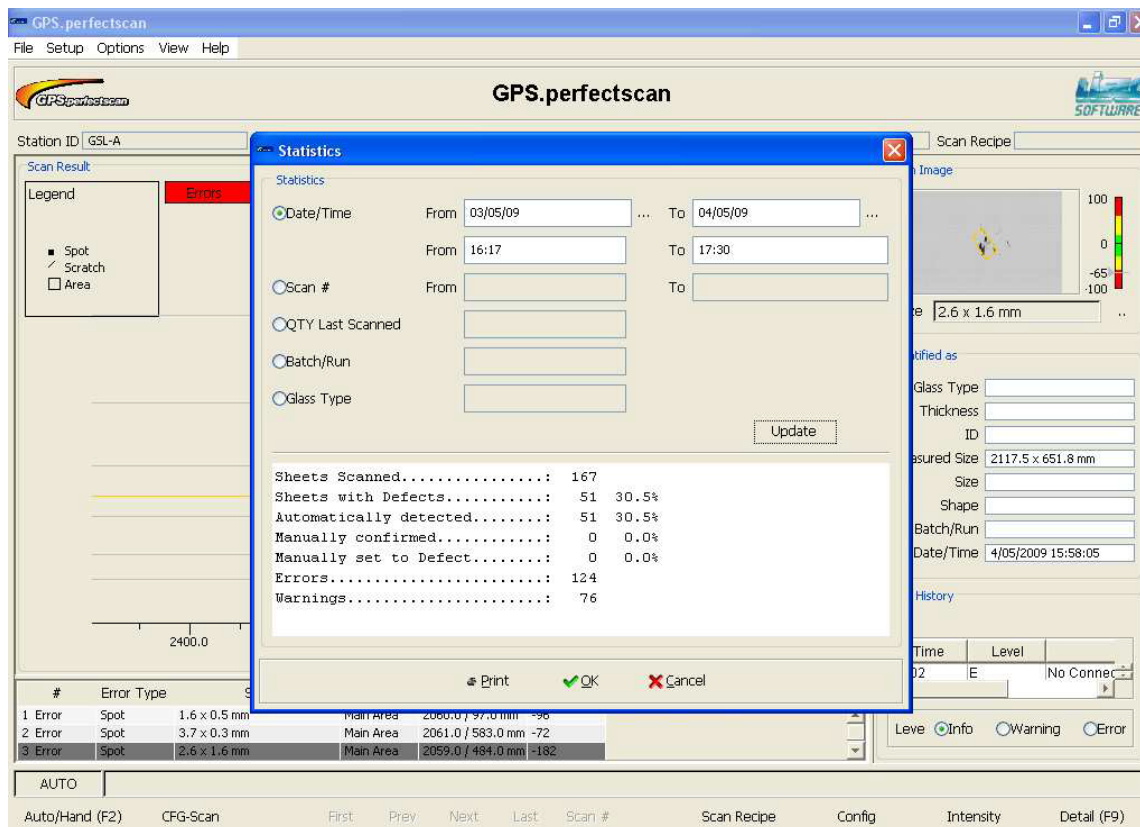
Il controllo di qualità ha certamente come obiettivo finale determinare se un prodotto ha una qualità conforme al mercato considerando le norme e le richieste di qualità del cliente finale. Questo è un sicuramente un punto fondamentale, ma non l'unico.

Un sistema di qualità va inteso anche come lo strumento fondamentale per poter intervenire sui processi produttivi al fine di identificare le cause della non conformità e prendere le iniziative di correzione degli stessi

Ad esempio se i report ci forniscono che un particolare difetto si presenta sempre sulla stessa posizione, molto probabilmente un macchinario potrebbe generare tale difetto.

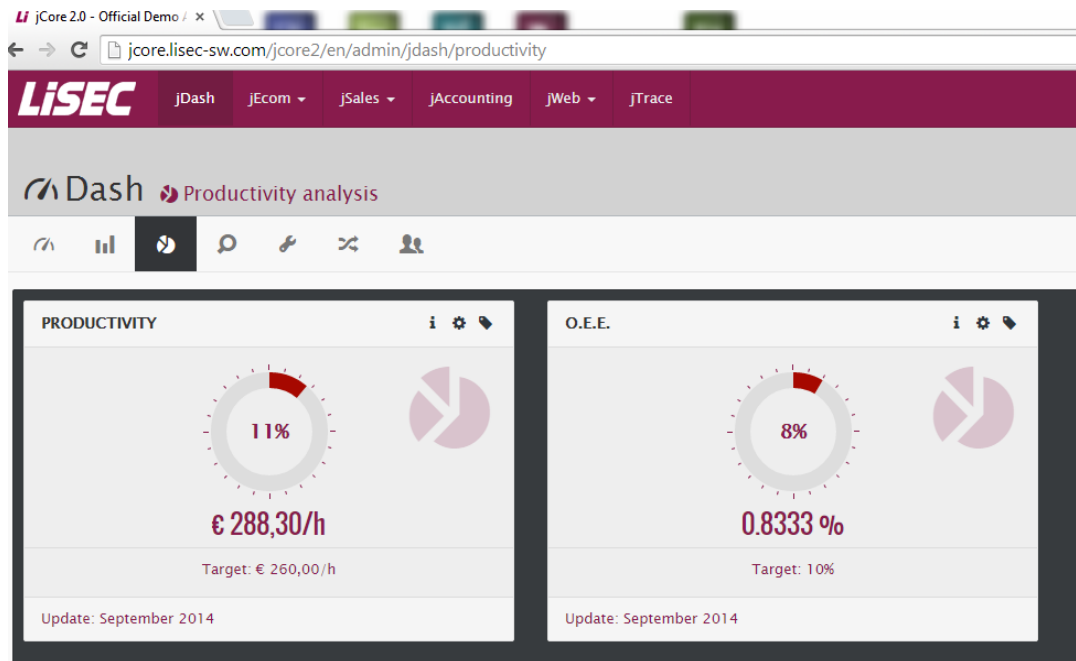
Un altro esempio potrebbe essere il caso in cui un difetto si presentasse soprattutto durante alcune ore del giorno probabilmente dovremo intervenire sul personale in turno.

Un sistema informatico legato alla qualità deve quindi essere in grado di fornire a posteriori anche i dati relativi all'intero processo produttivo



Fruibilità via web

Al giorno d'oggi ogni azienda necessita di controllare la propria produzione da remoto e richiedono di vedere lo stato della produzione in tempo reale ovunque la persona si ritrovi, a tale scopo è stata pensata la possibilità di visionare lo stato, le statistiche il risultato delle scansioni via web.



Tecnologie di sviluppo software

Al fine di sviluppare il prodotto oggetto del presente capitolato, si raccomanda l'uso delle seguenti tecnologie:

Software Client-base

Linguaggio di programmazione: C++ / php / javascript

IDE di sviluppo: Qt editor

Database relazionale: MySQL

La versione del software stand-alone, deve essere sviluppata in C++ con l'ausilio del Editor Qt. devono essere utilizzate le librerie standard dell'editor, ma saranno ben accettate eventuali nuove librerie ad-hoc sviluppate dal committente.

Il database, deve essere di tipo relazionale, ed è consigliabile MySQL. Può essere preso in considerazione PostgreSQL;

L'interfaccia web del prodotto, deve essere sviluppata in php con l'ausilio di javascript.

L'obiettivo è la realizzazione di un interfaccia APP-oriented, di conseguenza "responsive" per tutti i dispositivi mobili (smartphone e tablet), è suggerito l'uso del framework AngularJs (javascript), e del framework Bootstrap (CSS/LESS).

Licenza/uso del prodotto

Si intende che gli sviluppatori del prodotto oggetto di questo capitolato ne ritengono il copyright intellettuale e materiale.

Per consentire la diffusione, maturazione ed eventualmente l'uso del risultato, si richiede l'adesione alla licenza MIT (<http://opensource.org/licenses/MIT>)."