

¿Cuál es el sistema adecuado para mí?

Síntesis

Actualmente, el software de visión industrial se clasifica en dos líneas: la librería de visión convencional y el entorno de desarrollo integrado específico para la visión (IDE). Establecer cuál es el software apropiado para cada proyecto de visión depende de varios factores: la facilidad de uso, productividad, flexibilidad, el rendimiento, la integridad y el mantenimiento. El papel blanco utiliza estos factores para contrastar los dos enfoques de desarrollo de software y establece claramente las ventajas y los inconvenientes de cada uno. Este debate asume que las herramientas de visión disponibles en ambos tipos de software son similares (por no decir, idénticas) y no explora las posibles discrepancias entre estas dos herramientas. Asimismo, el debate ignora la plataforma de hardware que emplean las aplicaciones de visión para decantarse por una u otra.

Facilidad de uso

Desarrollar una aplicación utilizando una librería de visión exige tener conocimientos (algunos dirían conocimientos exhaustivos) y experiencia con lenguajes de programación tradicional como C/C++, C# o Visual Basic. También será importante estar familiarizados con las herramientas de desarrollo asociadas: editor de código, compilador, enlace y depurador. No obstante, como afirman muchos expertos del sector, adquirir y mantener este conjunto de conocimientos puede ser complicado y costoso. Por el contrario, trabajar con un IDE específico de visión requiere un conocimiento rudimentario de los principios de programación: control del flujo, variables y expresiones condicionales/lógicas. El conjunto de conocimientos mínimos exigidos hace que el IDE específico de visión sea accesible para una audiencia mucho más técnica.

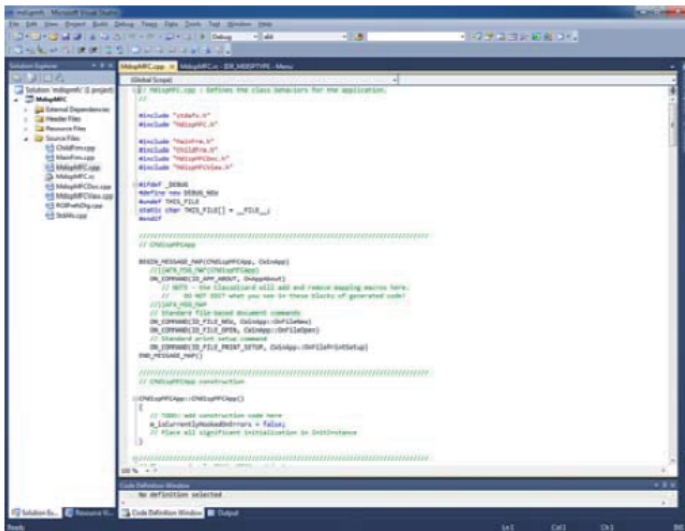


Imagen 1 - Desarrollar una aplicación con el uso de una librería de visión mediante la redacción de un código de programación tradicional (izquierda). Crear una aplicación con un IDE específico de visión conectando y configurando bloques de operación de configuración (derecha).

Productividad

El tiempo que tardemos en poder trabajar de forma productiva con una librería de visión dependerá en gran medida de nuestro conocimiento y experiencia en programación tradicional, así como la calidad y la capacidad intuitiva de la interfaz de programación de la librería de visión (API) y su documentación. Realizar un uso adecuado de una librería de visión exige un estudio exhaustivo de los ejemplos y la documentación de programación suministrados. Además, resulta sumamente beneficioso aprovechar las diferentes opciones de formación ofrecidas por las marcas de software antes de comenzar el desarrollo de la aplicación. Asimismo, deberemos invertir el tiempo necesario para diseñar adecuadamente la arquitectura del programa de aplicación inicial, ya que resulta esencial para su reutilización efectiva en proyectos sucesivos. Trabajar con una librería de visión suele conllevar plazos de desarrollo de semanas o meses.

Al contrario que la librería de visión, un IDE específico de visión se diseña para agrupar rápidamente y configurar el conjunto de operaciones que se necesitarán para una aplicación de visión típica: obtener la imagen siguiente, localizar (un) objeto(s), o (una) característica(s) de interés, analizar/medir/leer/descodificar, realizar

decisiones de aprobación/denegación y comunicar los resultados. La simplicidad de este enfoque hace que comenzar un proyecto nuevo (incluso desde cero) resulte sencillo. El automatismo de los requisitos de aplicación habituales (fijar una región de análisis en función de los resultados de una operación de localización) simplifica y, por tanto, acelera el desarrollo del proyecto. Y, la modificación de la aplicación en un centro de desarrollo es menos complicada por la naturaleza integral del entorno de desarrollo del software. Trabajar un IDE específico de visión exige, normalmente, unos plazos de desarrollo de días o semanas.

Flexibilidad

La librería de visión ofrece la mayor flexibilidad para gestionar aplicaciones que requieren tomar decisiones complejas importantes, el uso sustancial de una visión personalizada para otros algoritmos (como aprendizaje de matemáticas o maquinaria) junto con herramientas de visión preparadas y la necesidad de consolidar y operar múltiples visiones desde múltiples cámaras. Como ya se ha mencionado en la sección anterior, un IDE específico de visión resulta más apropiado para aplicaciones que respetan el modelo de uso previsto. Desviarse del modelo de uso previsto puede resultar complicado y conflictivo. Asimismo, añadir la visión personalizada u otras rutinas básicamente requiere una programación tradicional.

Rendimiento

Una librería de visión ofrece invariablemente el mejor resultado porque opera a un nivel más cercano al hardware. De hecho, un IDE específico de visión realiza por sí mismo una librería de visión de una forma u otra. Trabajar con una librería ofrece también más oportunidades de optimizar el rendimiento, incluyendo la paralelización manual de tareas y la descarga y permite el uso más eficiente de la memoria y la reutilización de recursos informáticos. Un IDE específico de visión ofrece un rendimiento inherente elevado pero la magnitud depende de la calidad de la implementación. Y normalmente, el uso de la memoria no resulta óptimo porque el IDE tiene que mantener la flexibilidad

Integralidad

Cuando se opta por usar una librería de visión, la aplicación de otras funciones de aplicación (es decir, la interfaz del operador y la comunicación con automatización y equipos empresariales) exige unaprogramación adicional personalizada o basada en librerías de terceros. Con un IDE específico de visión, la configuración de la funcionalidad auxiliar habitual (es decir, la interfaz del operador y la comunicación externa) es una característica clave del IDE. No obstante, las características avanzadas de visión se ocultan a propósito o no se exponen para garantizar la simplicidad y la facilidad de uso.

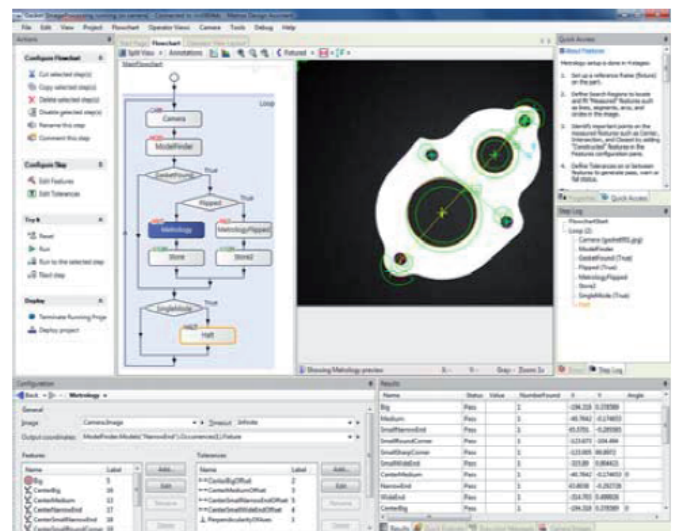
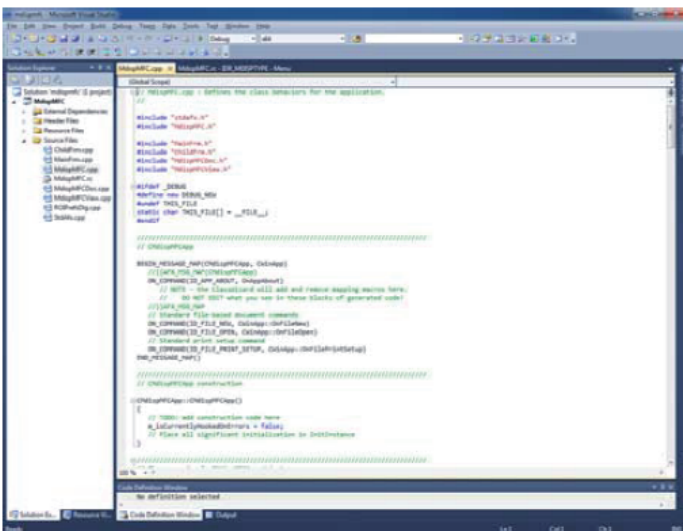


Imagen 2 – Crear una interfaz del operador al utilizar una librería de visión requiere el uso de herramientas independientes y una programación adicional (izquierda). Un IDE de visión específica integra la facilidad para crear una interfaz del operador a partir de una plantilla preconfigurada.

Mantenimiento

Una vez que se desarrolla una aplicación a través de una librería de visión, cualquier otro esfuerzo necesario para revisarla o adaptarla puede depender de forma sustancial de su complejidad, así como la calidad de su implementación y documentación. Asimismo, transferir esta responsabilidad a otro programador puede resultar un proceso lento y complejo. No es como un proyecto desarrollado con un IDE específico de visión, que es más fácil de transferir o compartir.

El veredicto

Elegir entre una librería de visión, como Matrox Imaging Library (MIL), o un IDE específico de visión como Matrox Design Assistant, depende de las circunstancias y los objetivos de aplicación. Quienes quieran y puedan invertir en obtener y mantener una técnica de programación tradicional y necesitan que su sistema de visión industrial ofrezca unos niveles de rendimiento y funcionalidades sin precedentes, no se equivocarán si usan una librería de visión. Los usuarios típicos de librerías de visión son fabricantes de equipos originales (EOM) que integran la visión industrial en maquinaria general para su comercialización en cantidades significativas a lo largo de muchos años. Por otro lado, quienes quieran pasar de forma rápida y habitual de un proyecto de visión industrial a otro, ofreciendo los niveles actuales de rendimiento y capacidad, lo más adecuado es un IDE específico de visión. Los usuarios de IDEs específicos de visión suelen ser integradores de sistema con personal técnico multidisciplinar que confluye en instalaciones puntuales o proyectos con un número reducido de instalaciones duplicadas. Algunas marcas comerciales de software de visión industrial, como Matrox Imaging, comprenden las diferentes necesidades y ofrecen productos que se ajustan a ambos tipos de usuario.

