White Paper

Myth and Reality of Auto-Correction in File-based Workflows

Bibhas Kumar Samanta & Asish Kumar Das Bera
Interra Systems, Inc.

INTRODUCTION

File-based workflows are ubiquitous in the broadcast world today. The file-based flow has brought enormous efficiencies and made adoption of emerging technologies like Adaptive Bit-Rate (ABR), 4K, UHD, and beyond possible. Multiple delivery formats are now possible because of file-based workflows and its integration with traditional IT infrastructure. However, the adoption of file-based flows comes with its own set of challenges. The first one, of course, is - does my file have the right media, in the right format and without artifacts? Fortunately, the leading auto QC tools have kept pace with the growing technology advances to provide us with this peace of mind. However, there are still many unsolved video artifact issues that the auto QC tools grapple with. Video dropout, for instance, is still a subject of research and significant advances are expected over the next few years to more accurately detect these issues. Once we know the issues, a natural question is – if the auto QC tool can detect the problem, can it also fix it? The answer to this is not so straightforward. In the analog and tape world, the content was as good as it was created. Correction was limited to simpler processes like signal level clipping, or color phase correction, which could all be done at the delivery stage. Of course if the content deteriorated due to tape issues like tears, twists   etc., one went through expensive film restoration techniques, if one could afford them – these were manually assisted processes done under the fine eye of the editor. Can auto-correction take care of these?

It so turns out that correction is not that simple in the file-based world. The content is often stored and delivered in a compressed format. It is also wrapped in containers to keep the audio, video, sub-titles and a host of metadata information for the tools to work properly in the workflow. In the file-based workflow, correction of the content requires not only changes to the baseband content, but also re-encoding and re-wrapping of the corrected content back to the compressed format.

We see the following challenges to the auto correction process:

- Firstly, there are several baseband issues that are not even detected automatically (in other words, they are outside the scope of auto QC), forget about auto correcting them. Remember, analog world people used manually assisted processes under editorial supervision.

- Secondly, after the corrections are applied, through manual or automated process, if the transcode including the re-wrap processes are not managed properly, auto correction will introduce new set of issues – the corrected content may even be worse than what you started with, resulting in an unproductive looping.

How then can you depend upon an auto QC tool to do auto-correction? Well, there are some issues that are amenable to auto-correction, albeit a few. Most of the issues can be categorized in three types: metadata inconsistency, video essence issues, and audio essence issues. Whenever re-encoding and re-wrapping of content is required to be done after correction, auto-correction via auto QC tool may not work very well. On the other hand, metadata inconsistencies and audio essence issues are more amenable to auto-correction. However, correction of video essence issues typically do not converge when performed on encoded files. In fact some of the video issues take place due to encoding / transcoding process as well. The right workflow, tools and techniques are needed to be deployed to make the auto-correction flow work well for you.

It is a misconception that auto-QC tools can also auto-correct all kinds of content issues. That is a sweeping generalization, and QC tools should not fuel that false notion. This paper attempts to clear the misconception and also sheds light on the extent to which auto-QC tools and other tools in the workflow can auto-correct issues in media content.

Figure 1 below, provides a typical high level file-based workflow. After ingest, content is edited to create a mezzanine file which is of high quality but with minimal compression. Different facilities can select their own mezzanine format ranging from Motion JPEG to ProRES to AVC Intra. Mezzanine content is then transcoded to multiple compressed formats for different delivery formats.

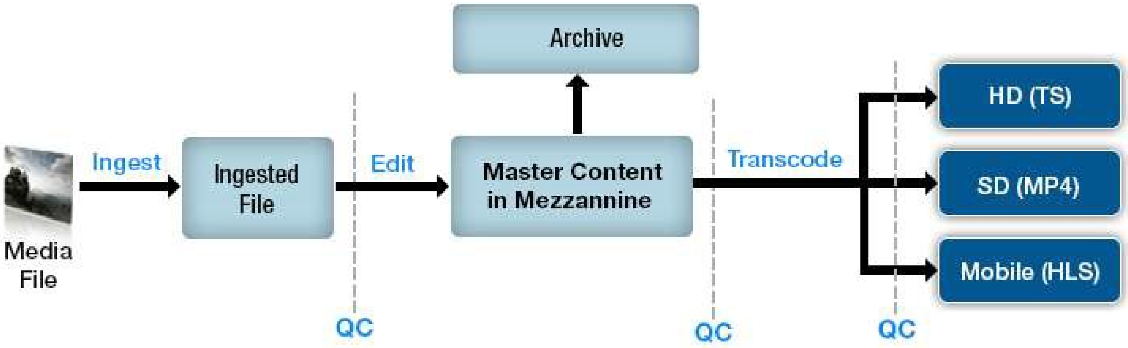

Fig. 1 - Typical File-based Workflow

Since content undergoes some form of complex transformation at each stage, this can potentially introduce stage specific issues within the content. Each stage can introduce different types of issues. Similarly, different levels of correction are possible at different stages. Auto-correction works well if it is done on uncompressed digital content, as it can be modified and corrected, before being compressed and wrapped, much like the correction in the analog world. However, if the content is already encoded and wrapped (e.g., transcoded content), then auto-correction gets far more complicated - the re-encode process after the auto-correction introduces other issues, making the correction process divergent, less effective and even infeasible.

The ingest process often introduces artifacts like dropouts, signal level errors in video and transient noise, wow and flutter in audio. Auto-correction can work well at this stage if the digitized uncompressed and un-wrapped content is available. Issues like video signal levels, RGB color gamut, audio loudness etc., can be corrected in the uncompressed digital content, which is then encoded and wrapped into the mezzanine format like AVC intra and J2K. However, there are a host of baseband video and audio issues which should not be auto-corrected as one runs the risk of modifying the content.

The ingest process often introduces artifacts like dropouts, signal level errors in video and transient noise, wow and flutter in audio. Issues like video signal levels, RGB color gamut, audio loudness etc., can be corrected in the uncompressed digital content, which is then encoded and wrapped into the mezzanine format like AVC intra and J2K. Although, auto correction will work properly for most of the error scenarios, however, the correction may modify the content to an unacceptable level. For example, while correcting VSL or RGB color gamut errors, the characteristics such as hue, saturation or contrast might be changed affecting the perceptual experience of the viewer. Similar example holds true for correction of transient noise in case of audio. In these cases, manual inspection is also required after correction of the content.

The similar argument holds for the editing stage. However, there are several other types of issues that can crop up at the editing stage which cannot be auto-corrected. Trying to merge two different media during the editing process can lead to field order issues. We often see customers complaining about VSL, RGB errors which get introduced at the editing stage while adding special effects and graphics/text in the content.

After transcoding, auto-correction can be done to a very limited extent. Transcoding for delivery purposes is a complex transformation where content is converted from one format to other. Many issues such as audio / video corruption, blockiness, blurriness, pixelation, audio / video dropouts, motion jerks, audio clipping etc. have been found to occur during the conversion process, not to forget non-compliance with audio/video formats or delivery specifications. Transcoders can also get affected by buffering issues during transcoding process, leading to overflow/underflow like situations. This can lead to introduction of freeze or silence frames within the content. Even if we were to auto-correct the issues, re-encoding the corrected content has the potential of introducing similar issues in different forms and different parts of the content. It is the best that auto-QC and transcoding tools collaborate to correct these issues.

With this background, let's now have a look at different kinds of issues that an auto QC solution can detect in an encoded content and what needs to be done to correct those issues.

Any QC solution will typically detect four kinds of issues:

- **Conformance Errors:** These errors are primarily non-compliance to different audio video standards. For example, an MPEG-2 video stream must be compliant with the MPEG-2 video standard. Any non-conformance needs to get reported. This category also includes checking compliance of the content against different regional/delivery specifications like DPP, IMF, AS-02 etc. Correction of these kinds of issues generally requires the files to be re-encoded and re-wrapped. Baseband correction is not required for these kinds of errors.

- **Metadata Errors:** Each workflow and each stage in the workflow has its own requirement in terms of metadata. For example, an HD delivery requires resolution to be 1920 x1080. Content meant for broadcasting in the USA needs to have a frame rate of 29.97 fps. Each delivery or stage can have further restrictions on parameters like scanning type, GOP structure, profile and level of encoded media, the number of audio / video tracks etc. Any deviation from the acceptable values will lead to content being rejected. So a QC solution is expected to check for such metadata properties at different stages of the workflow. Moreover, certain information like resolution, time-codes and field order are present / encoded at both the wrapper and at the audio/video level. If there is any inconsistency between the layers, a QC solution should be able to report the same. If the issue is with the wrapper layer (MXF, QuickTime, Transport) then only re-wrapping needs to be done to correct the content. But in cases, where metadata information at video/audio level is incorrect, one will need to perform basic re-encoding along with re-wrapping. Example for such a case would be if the US media environment requires content with 29.97 fps but underlying media has frame rate of 24 fps. Simple fix for this issue would be to introduce cadence pattern of 3:2 at video layer. Such correction will need basic modifications to video layer and further re-wrapping of compressed media.

- **Baseband Errors:** These errors are different audio / video artifacts which lead to deterioration in perceivable quality of content. These errors are introduced because of stage specific transformations as discussed earlier. This includes errors like freeze frames, blockiness, dropouts in video and issues like silence, different kinds of noises in audio. Correction of such artifacts first needs to be done at the baseband level followed by the re-encode and re-wrap processes.

- **Regulatory Compliance Errors:** Different regions of the world have their own regulations in terms of content quality. It includes loudness control regulation all over the world. We have CALM Act in the USA, EBU R128 standard is widely followed in Europe. Likewise, the UK broadcasting market requires content to be checked for any possible flash patterns to avoid photo sensitive epilepsy situations. It is possible to correct these kinds of errors via baseband correction followed by transcoding process.

Many kinds of errors discussed above will require compressed content to be processed (re-encoded or re-wrapped or both) to remove the errors. This processing is not as straightforward as it looks. It is critical to decide where in the workflow, and with what tools these errors should be corrected. Let's take the case of an auto QC tool also claiming to provide "good" correction capabilities.

The tool comes with its own encoder. The workflow would look like this:

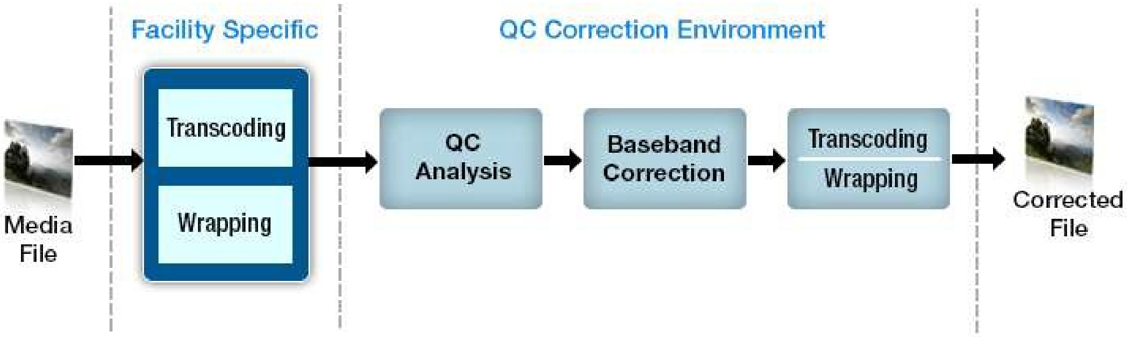

Fig. 2 – QC Tool-based Transcode & Correction Workflow

So a typical flow for auto QC and correction flow will work somewhat as below:

- Mezzanine file is converted to delivery format using facility specific transcoder
- The transcoded content is then checked using a QC tool and an output report is generated; The report will contain detected errors, if any
- If the content has no errors, it goes to the play-out stage otherwise it moves to correction workflow which is an extension of the QC tool here
- The QC tool then performs correction on the basis of the reported errors - it uses baseband correction algorithms along with its transcoder for correction of the content

The corrected content is then ready and can be moved to the play-out stage for final delivery. Up to this point everything looks good and quite rosy. But users of this type of workflows may be in for a shock when corrected content fails to meet the delivery requirements and gets rejected. The situation is quite common because the corrected content may not be of desired quality and may have additional new issues which were not there in the first place. Let's now look at the challenges involved in the above correction process.

TRANSCODING

Transcoding is a complex transformation process involving conversion of content from one form to another. A transcoder output is controlled based on a host of input settings to handle varying flavors of container and media formats, and to meet various kinds of delivery specifications, in order to get the media with a required level of quality and so on. The input settings control various internal processes of the transcoder which includes motion estimation techniques, bit budgeting, rate distortion model, selection of QP values and matrices, the block interpolation/estimation processes, reference frame selection and more. The final output of the content is dependent on the quality of the said processes being used inside the transcoder as well as the input parameters selected. Inappropriate selection and usage of input settings to transcoder may result in output content not meeting the intended requirements. A wrongly selected bitrate parameter can degrade the quality of the output content with new artifacts (out of RGB color gamut errors, video signal level errors, blockiness, softness etc.). Another such scenario can come up while selecting display field order for the output content. An SD DV content (bottom-field first by default) when transcoded to MPEG-2 video (top-field first by default) will lead to motion-judder issues in the output because it was required to change the default field order input value to the required one. Thus in order to create good quality and optimally compressed content, several parameters need to be fine-tuned and managed as per the facility's requirement. Setting these parameters/options even for the best transcoder requires expertise. One cannot expect another 'generic' transcoder to be able to perform at the same level. It is hence to be expected that any attempt to re-encode the content with another encoder could lead to negative effects. The second encoder, while trying to encode the corrected content at the same bitrate may follow a different bit allocation strategy leading to compression issues like blockiness etc. It is also highly possible that a new encode process can completely miss certain information that is vital for the content. To name a few, user data present at video level may get lost in the process of transcoding. Another example would be watermarks, where the generator leaves a special mark in the video/audio to establish publisher information. It is impossible to replicate or reinsert these watermarks unless the same set of tools is used during correction. It is also quite possible that some of the settings are not even consistent among the two transcoders. For example, the other transcoder might be using different motion estimation techniques or rate distortion algorithm inside it or it may also happen that the original set of tools inside a profile or level is not supported. That will cause the correction process to generate media data with unacceptable profile / level and content quality, which will be rejected later at the play-out stage. At a minimum, one should use the same encode process and tools as used during content creation.

However, issues don't end here. Even if we plan to use the same transcoder, it can potentially introduce new errors while correcting existing ones.

- The re-encoding process leads to loss of some audio / video information which in turn impacts the quality of content. Degradation in quality, though minimal for most of the cases, will depend on the encoding parameters and the content itself. If re-encoding is done to reduce bitrate of the content, it will lead to compression artifacts like blockiness, pixelation etc.
- Conformance errors may also get introduced because of faults in encoder under certain conditions.
- In a few cases, it is also possible that metadata errors may be introduced, if the wrapper information is not set correctly; one such example could be the field order - assume a case where the field order has changed after re-encoding but the same is not reflected at the wrapper level. Such inconsistencies can arise and thus there is a need for a better management of such issues.

RE-WRAP

Another big challenge in a correction flow is to re-assemble / re-wrap the corrected and compressed media with exactly the same properties as the original file. Transcoders come with their own built-in Muxers or can be integrated with third party Muxers to wrap compressed media into a container. The media workflows in the broadcast industry use their own set of unique tools to transform and assemble media information. A different re-wrap tool or the same tool with differently encoded essence will produce different results. This implies that the corrected output file may be different in properties in comparison to the input source file. An example is MXF version, where the original file may have been assembled using a lower version. But if a new Muxer used during the correction process uses a higher MXF version, it may cause interoperability issues in the workflow. Also, the MXF specification allows addition of new proprietary ULs that can be generated and interpreted by specific Muxers / applications. For other tools, it acts as '*Dark Metadata*' that will be ignored while processing. Hence, the second Muxer for such cases will ignore the dark metadata and the proprietary information would be missed in the corrected content. Hence, it's an imperative to avoid the usage of two different Muxers in your correction workflow.

BASEBAND CORRECTION

In baseband correction, there are issues like video signal level, RGB gamut, field order, digital dropout, loudness related errors which can be intelligently corrected. For such issues, the content is first decoded. Algorithms are then applied on the baseband / uncompressed signal to intelligently correct them for the reported issues. Once the baseband correction is done, the content is re-encoded and re-wrapped. Can we fully rely on baseband correction? Perhaps not. It is possible that a certain correction may introduce fresh errors during re-encoding process. For example, VSL / RGB correction may end up altering the block boundary pixels which in turn leads to blockiness like issues in the corrected content. There are additional set of errors which cannot be auto corrected like: freeze frames, silence and certain noises. If the capture device, for some unknown reasons, fails to capture a few frames, it can potentially lead to a freeze like situation. It's not possible to re-create those dropped frames during correction cycle until we have access to the dropped frames. It is also possible that certain special effects that are added to video may cause QC solution to detect those effects as blurred or pixilated area in the video frame. In this scenario, it is not desired to correct the content. Hence, there is a need for manual intervention to understand these anomalies and then take appropriate corrective measures. Some of these aberrations maybe intentionally be introduced as special effects, and therefore, needs no correction.

PROPOSED WORKFLOW

Auto correction has its own set of challenges as mentioned in the previous section. Because of these challenges, it is not practical to expect an auto-correction tool to be a panacea for all issues. In fact, there is a class of issues that can be auto-corrected. Coupled with the right set of tools and workflow, one can make auto-correction work under these limited circumstances, such as:

1. ***Legalization of audio and video content and some cases of regulatory compliances.*** These include audio loudness, true peak, loudness range, audio levels, audio noises like background noise, crackle. On the video side, it includes video signal levels, RGB color gamut, cases of video dropouts and also flashiness patterns. The proposal here is to limit the role of QC solution to baseband correction. The correction flow can then rely on facility specific transcoder for its encoding needs. For example, a facility may depend on Dolby tools for encoding of AC-3 /Dolby-E content. In such a scenario, the role of auto QC tool is to perform baseband correction for audio and then submit the encoding job to Dolby tools. This would ensure consistency between the original content and the corrected content in terms of metadata and quality.

Another practical use case here would be integration of auto QC tools with the workflow automation / transcoding solutions like Telestream Vantage. Once the content is transcoded, QC tools can then perform content analysis followed by intelligent correction depending on the detected and correctable errors. The workflows can be further configured to feed the corrected but uncompressed output from QC tools to in house transcoders like Vantage for re-wrapping/re-encoding. Submission of transcoding jobs to the transcoder after correction can be initiated in multiple ways. In some cases it can be as simple as dropping a file in a Watch Folder while for other cases, a QC solution may need to invoke transcoder's web services to start the required job. For larger workflows, it would also make sense for MAM /workflow automation solutions to create some kind of correction / self-healing workflows so that transcoding action can be invoked once correction process is done. These discussed approaches would require the QC solution to be integrated with some of the widely used transcoders / workflow solutions so that a large number of customers get the benefit. Such a flow would typically look like this.

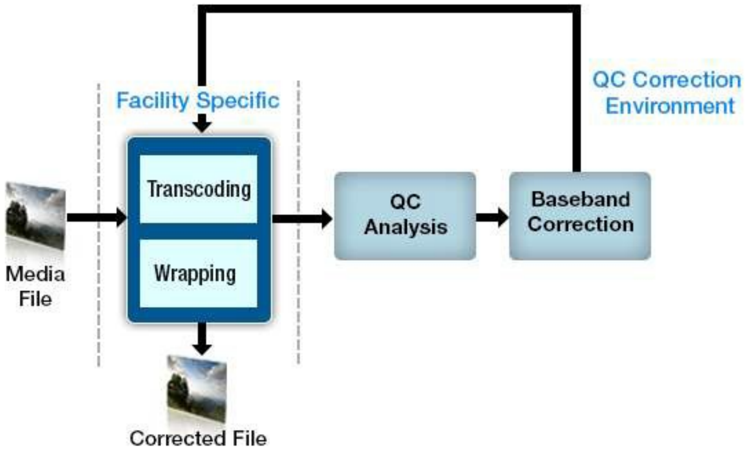

Fig. 3 - Proposed Transcode & Correction Workflow

The steps followed in the proposed flow are listed below:

- Media file is analyzed using the QC tool
- The content moves to the play-out folder if it passes. In case of a failure, the content is de-muxed, decoded and then corrected for anomalies at the baseband layer, if required
- Content is then submitted for transcoding using facility specific tools
- The correction process may also need to specify new parameters / settings to be used during the transcoding stage
- Modes for submission of transcoding job can vary as discussed earlier

The use of the same transcoder will eliminate lots of potential issues and make the above flow more practical and amenable to correction.

Most of the challenges in correction process arise because of re-encoding / re-wrapping. Correction which does not change the size of the compressed content can be handled without a possible re-wrap. That is true for uncompressed content based on baseband correction. Audio content in a lot of cases is stored in an uncompressed manner using formats like PCM, AIFF, BWF or AES3, owing to the fact that audio requires much less data size as compared to video. Since uncompressed content occupies fixed block sizes at certain offsets, it is not necessary to re-wrap the whole media. A smart correction tool can simply perform what we call as **in-place correction.** The goal here is to un-wrap the audio, record the length of each uncompressed audio block with the corresponding file offsets. Once baseband correction has happened, corrected content can then be written back to the main file block by block using the recorded information. This way wrapper information or media data from other tracks remain untouched.

The above strategy is really useful for correcting audio errors like program loudness, loudness range, true peak etc. and it works efficiently in an iterative correction process. Errors like loudness, loudness range cannot be corrected in a single run. They may require multiple correction runs to reach desired levels. In-place correction ensures that no temporary file or buffer needs to be maintained for storing intermediate media. Corrected output values can be re-written to the final file for each iteration. This strategy works out not only to be efficient but also fast. The concept of in-place correction can also be extended to uncompressed video formats like YUV, RGB. But since uncompressed video formats are not widely used, it may not be very beneficial to the end customer.

2. ***Another class of issues that can be corrected is metadata inconsistency errors.*** In cases where the encoded content is correct, but container metadata has been wrongly encoded, the problem of correcting the content requires only metadata changes for specific fields. These corrections can be applied without the need for transcoding or rewrap of the content and are very amenable to auto-correction. This again falls into the category of in-place correction. For -example, if there is a discrepancy between the resolution information present at the MXF layer compared to the actual video resolution, the resolution information at the MXF layer needs to be corrected by directly accessing the headers and there is no need to transcode or re-wrap the file. This scenario would include correction of metadata fields like frame rate, chroma format, aspect ratios, sampling frequency, encoded duration etc.

PRODUCT BRIEF: BATON AND BATON CONTENT CORRECTOR

Baton is an enterprise-class QC or automated content verification solution for file-based media used by global telcos, broadcasters, post production houses, IPTV and archiving companies. Baton is the trusted choice for their file-based QC needs with its comprehensive quality checks, scalability, and support for wide range of media formats, and an intuitive web-based interface.

Baton comes with an add-on utility called Baton Content Corrector (BCC). BCC has the ability to auto-correct a host of audio, video and metadata errors detected by Baton in the media files. This includes correction of errors like program loudness, loudness range, true peak, video signal levels, RGB gamut and a range of metadata corrections. In-place correction as discussed earlier is used for correction of uncompressed audio content. BCC is seamlessly integrated with Baton via Smart Folders. Users of Baton/BCC can configure Smart Folders to submit jobs for correction in case any correctable error is reported. BCC also has support for web services in case it needs to be invoked remotely.

CONCLUSION

Auto QC is now an essential component in file based workflows and is widely used these days. This has triggered the need for a QC solution which can auto-correct errors in order to save time and resources. It is based on the thought that if a tool can detect error, it can also potentially fix it. But auto-correction in the file-based world is a more complex process and should not be trivialized. A QC tool having in-built support for auto correction including transcoding has issues of its own. Transcoding and re-wrapping processes if not managed properly, can introduce fresh issues into corrected content leading to further degradation of content quality. Hence, it is not possible to fully rely on such auto correction flows. A more practical approach would be to reuse facility specific tools for encoding needs during the correction process. In such scenarios, the role of a QC tool is limited to baseband and metadata correction or setting the transcoder correctly. A smarter in-place correction strategy can also be adopted in case of uncompressed content. Having said this, there is still a set of issues which requires manual intervention and thus cannot be auto corrected. Hence, the scope of QC tools for auto correction is limited but feasible for a set of issues provided we use the right tools, workflows and techniques.

The Auto QC tools like Interra Systems' Baton are used to automatically detect the video and audio artifacts. In recognition of the above facts, Baton focuses on legitimate auto-correction in a controlled and restricted manner.

ABOUT INTERRA SYSTEMS

Interra Systems is a leading global provider of software products and solutions for quality control, real-time monitoring and video analysis to the digital media industry. The company's solutions include Baton, an automated verification system that ensures media content readiness, Vega, a family of audio/video analyzers that accelerate media product development and Orion, a real-time content monitoring solution. Interra Systems is headquartered in Cupertino, CA. For more information, please visit http://www.interrasystems.com.

Interra Systems, Inc.
1601 S. De Anza Boulevard, Suite 212
Cupertino, CA 95014
USA