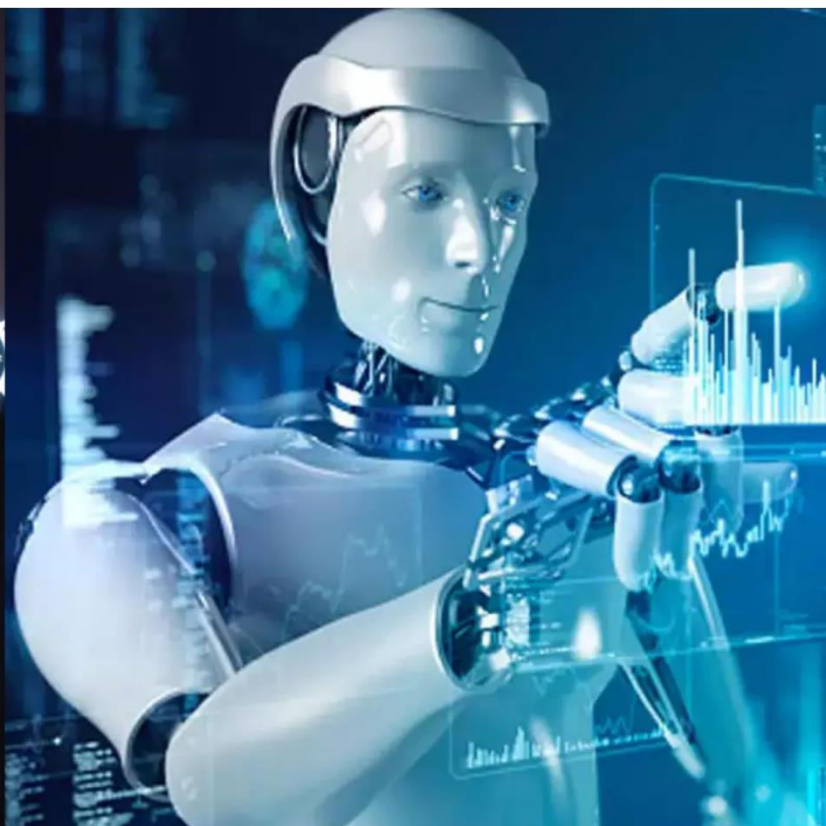




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 8, August 2026

A Review of Media-to-3D Reconstruction Techniques using Gaussian Splatting

Dr. Namrata Khade¹, Mayur Raju Nandanwar², Shivraj Shreeram Mahajan³,

Lalit Arunrao Wanjari⁴, Massom Patel⁵, Priyanshu Ramsamujh Yadav⁶

Department of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, India¹⁻⁶

ABSTRACT: Turning ordinary photographs or video into a usable 3D scene has been a core problem in computer vision for over two decades, and the tools used to solve it have changed dramatically in just the last few years. Classical pipelines built around Structure-from-Motion (SfM) and Multi-View Stereo (MVS) are still the backbone of most reconstruction systems — they recover camera positions and dense geometry reliably, but the output is not naturally suited for fast, photorealistic rendering. Neural Radiance Fields (NeRF) changed the conversation by treating a scene as a continuous function that could be queried along camera rays, producing remarkably realistic novel views, though usually at a steep cost in training and rendering time. More recently, 3D Gaussian Splatting (3DGS) has offered a middle path: an explicit set of optimizable Gaussian primitives rendered through a fast, differentiable rasterizer, giving both strong visual quality and real-time performance. This review walks through five papers that, together, trace this evolution — from camera-geometry recovery, to dense depth estimation, to implicit neural fields, to explicit Gaussian representations, and finally to newer directions such as feed-forward Gaussian prediction, Gaussian-based SLAM, and mesh extraction from Gaussian scenes. For each paper we summarize the problem being addressed, the method proposed, what it achieved, and where it falls short. Bringing these threads together points to a clear gap in the literature: no single existing pipeline takes raw, unposed media all the way from ingestion to a validated, renderable, and optionally mesh-exportable Gaussian scene. This review closes by outlining that gap and the direction it motivates.

KEYWORDS: 3D Reconstruction, 3D Gaussian Splatting, Structure-from-Motion, Multi-View Stereo, Neural Radiance Fields, Novel View Synthesis, Differentiable Rendering, Media-to-3D Pipelines.

I. INTRODUCTION

Reconstructing a three-dimensional scene from a handful of photographs sounds almost simple until you try to do it. A camera collapses a 3D world into a flat, 2D image, throwing away depth information in the process, and a reconstruction system has to work backward from that loss — figuring out where each photo was taken from, how the different views relate to one another, and what shape and appearance the underlying scene must have had to produce exactly those images. For a long time, the standard answer to this problem was Structure-from-Motion paired with Multi-View Stereo: match features across images, recover camera poses, triangulate a sparse point cloud, then densify it. This combination is dependable and well understood, and it still forms the geometric backbone of most modern pipelines.

What SfM and MVS were never really built for, though, is rendering. Their output — camera poses, point clouds, sometimes a mesh — is a geometric record of a scene, not a representation optimized for producing new, convincing views of it. That gap is what Neural Radiance Fields addressed when they arrived: instead of storing geometry explicitly, NeRF trains a neural network to predict color and density at any point along any camera ray, and stitches novel views together by sampling that network repeatedly. The results were genuinely striking, but the approach carries real practical costs, since every rendered pixel requires many network evaluations, which makes both training and rendering slow.

3D Gaussian Splatting is where this review's attention eventually settles, because it reframes the trade-off entirely. Rather than hiding the scene inside network weights, 3DGS represents it explicitly as thousands (often millions) of small, anisotropic Gaussian "blobs," each with its own position, shape, opacity, and color, optimized directly against the training images and rendered through a fast rasterizer rather than ray marching. The result keeps much of NeRF's visual quality while running at real-time or near-real-time speeds.

The five papers reviewed here were chosen because each one addresses a different stage of the pipeline that a complete media-to-3D system needs — camera recovery, dense geometry, implicit and explicit rendering, generalizable Gaussian prediction, online/SLAM-style reconstruction, and mesh extraction from Gaussian scenes. Read individually, each paper solves a narrow, well-defined problem. Read together, they suggest something the literature has not yet fully delivered: an integrated pipeline that takes raw media in at one end and produces a validated, renderable, and — where needed — mesh-exportable 3D Gaussian scene at the other, without treating each stage as its own disconnected research project.

[1] Review of "VGGT: Visual Geometry Grounded Transformer" Authors: Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, David Novotny (CVPR 2025 — Best Paper Award)

For almost a decade, recovering camera poses and sparse structure from a set of images has meant running an incremental SfM pipeline — match features, verify geometry, register views one at a time, refine everything with bundle adjustment. It works, but it's inherently sequential and can be slow on large collections. VGGT [1] asks a more disruptive question: what if a single feed-forward network could just look at the images and output the camera parameters, depth maps, point maps, and point tracks directly, with no iterative optimization loop at all?

The answer, in this paper, is yes — and the results are hard to ignore. VGGT [1] is a large transformer trained end-to-end to take in anywhere from one image to hundreds of them and, in a single forward pass, predict:

- Camera extrinsics and intrinsics for every input view.
- Dense depth maps and 3D point maps for the scene.
- 3D point tracks across the input views, useful for downstream tasks like tracking or SLAM.

Unlike earlier learning-based pose regressors, VGGT [1] doesn't need a visual-geometry optimization pass afterward to clean up its predictions — the raw network output is already competitive with, and in several benchmarks better than, classical SfM pipelines that do use bundle adjustment. Reconstructing hundreds of images takes under a second on suitable hardware, compared to the minutes-to-hours a full incremental SfM run can take on the same collection.

This kind of feed-forward reconstruction is being used for:

- Fast camera-pose initialization for downstream novel-view synthesis pipelines, including 3D Gaussian Splatting — the authors note VGGT's outputs can be exported directly in COLMAP format for exactly this purpose.
- Dense point-cloud and depth estimation from unposed image collections, without a separate MVS stage.
- 3D point tracking, extending the same backbone to video-style tasks rather than just static reconstruction.
- Low-latency applications such as visual localization and robotics, where waiting for classical SfM to converge isn't practical.

What makes VGGT [1] compelling is that it collapses several traditionally separate stages — feature matching, pose estimation, dense depth, sparse-to-dense fusion — into one trained model, which is a genuinely different design philosophy from the classical pipeline this review has otherwise focused on. It suggests that camera and geometry estimation don't have to be a sequential, per-scene optimization process; a sufficiently well-trained network can amortize that cost across a huge variety of scenes seen during training.

The trade-offs are the ones you'd expect from a learned, feed-forward system. Accuracy, while strong, can still trail a fully converged bundle-adjusted reconstruction in high-precision settings, since there's no per-scene refinement loop correcting the network's initial guess. The model is also large and trained on broad data, so its behavior on scene types far outside its training distribution — extreme lighting, very thin structures, highly repetitive textures — is less predictable than a classical geometric pipeline whose failure modes are, at least, well understood. And because it's a black-box network rather than an explicit optimization, it's harder to plug in the kind of manual quality gates (reprojection-error thresholds, connectivity checks) that a classical SfM stage naturally exposes.

[2] Review of "DUST3R: Geometric 3D Vision Made Easy" Authors: Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, Jérôme Revaud (CVPR 2024)

Traditional MVS has always leaned on a hidden assumption: that camera intrinsics and extrinsics are already known, or at least separately estimated, before dense depth can be triangulated. Getting those calibration parameters is its own tedious sub-problem, and any error there propagates straight into the dense reconstruction. DUST3R's [2] contribution is to sidestep that dependency almost entirely by reframing dense 3D reconstruction as a direct regression problem, with no explicit camera model required as input.

The central idea is the "pointmap" — instead of predicting a depth value per pixel relative to a known camera, the network directly regresses a 3D point in a common reference frame for every pixel of every input image. Concretely, the method:

- Takes a pair of images and, through a transformer encoder-decoder built on strong pretrained backbones, regresses a dense pointmap for each image, both expressed in the same 3D frame.
- Recovers camera intrinsics, relative pose, and even absolute pose as a byproduct of the pointmap regression, rather than needing them as inputs — pixel matches and focal lengths can be read back out of the predicted geometry.
- Extends beyond pairs through a global alignment step, which stitches many pairwise pointmap predictions from a larger image collection into one consistent scene, without a separate bundle-adjustment stage in the classical sense.
- Unifies monocular and binocular reconstruction under one formulation, since a "pair" can just as easily be two views of very different overlap, including near-degenerate cases classical MVS struggles with.
- This kind of calibration-free dense reconstruction is used for:
- Dense multi-view stereo on uncalibrated, "in the wild" image collections where no camera metadata is available at all.
- Monocular and multi-view depth estimation, since the pointmap formulation handles both cases without switching architectures.
- Recovering relative and absolute camera poses directly from the predicted geometry, making it usable as a lightweight alternative to classical SfM for smaller image sets.
- As a geometric front end for later feed-forward systems — VGGT and several other 2024–2025 methods explicitly build on the pointmap idea DUST3R introduced.

What's genuinely useful about this approach for a media-to-3D pipeline is that it removes one of the most fragile links in the classical chain — the assumption that calibration is solved before geometry begins. For casually captured video or photo sets where EXIF data is missing or unreliable, this matters a lot. It also blurs the line between "sparse" and "dense" reconstruction, since the same network that gives you camera poses also hands back dense geometry, rather than treating these as two pipeline stages with separate tools.

The limitations mostly show up at scale. The pairwise formulation is naturally suited to a handful of images with strong pairwise overlap, and the global alignment step needed to merge many pairwise predictions into one consistent scene becomes more expensive and less precise as the image count grows into the hundreds — this is part of why follow-up work like VGGT [1] moved toward a single joint forward pass over all views rather than pairwise-then-align. Precision in well-calibrated, dense-coverage scenarios can also still trail a fully converged classical MVS pipeline, since there's no per-scene photometric refinement loop the way there is in cost-volume-based MVS networks. Even so, DUST3R [2] marked a genuine shift in how the field thinks about the SfM/MVS boundary : treating camera calibration as something to be recovered incidentally rather than assumed upfront is now a standard design pattern in the newest reconstruction pipelines.

[3] Review of "3D Gaussian Splatting for Real-Time Radiance Field Rendering"^[1] Authors: Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis (ACM Transactions on Graphics, 2023)

This is arguably the most consequential paper in this review, so it's worth examining carefully rather than treating it as a given. By the early 2020s, NeRF-style implicit fields [9] had proven that photorealistic novel-view synthesis from photos was achievable, but their reliance on per-pixel ray marching through a neural network made both training and rendering expensive — real-time playback of a trained scene was, for most implementations, out of reach. This paper's central move is to drop the implicit neural function entirely and represent the scene explicitly instead.

The scene is modeled as a set of 3D Gaussian primitives, each carrying:

- A 3D mean (position) and an anisotropic covariance, decomposed into a scale and rotation so it can be optimized with gradient descent while staying a valid covariance matrix.
- An opacity value controlling how much the Gaussian contributes when composited with others.
- View-dependent color, represented with spherical-harmonic coefficients, so appearance can shift realistically with viewing angle.
- Two things make this representation practical rather than just elegant:
- Initialization from a sparse SfM point cloud, giving the optimization a geometrically sensible starting point instead of random primitives scattered in space.
- Adaptive density control, which splits Gaussians in under-reconstructed regions and prunes ones that aren't contributing, so representation capacity grows and shrinks where the scene actually needs it during training.
- A custom, visibility-aware differentiable rasterizer that projects and alpha-composites the Gaussians directly onto the image plane, avoiding ray sampling altogether — this is the piece that unlocks real-time rendering.
- This approach is used for:
- Real-time novel-view synthesis and interactive scene exploration, reported at 1080p in real time on suitable hardware — a genuine step-change from NeRF-era rendering speeds.
- Fast training compared to classical NeRF, since the explicit representation and rasterization avoid the heaviest cost of implicit volumetric rendering.
- As the base representation for a large and fast-growing ecosystem of follow-up work — compression, large-scale scenes, dynamic scenes, SLAM, and mesh extraction have all been built directly on top of this formulation, several of which are reviewed later in this paper.

The strength of this paper is that it doesn't trade quality for speed so much as find a representation where both are achievable together — the explicit, rasterized approach turns out to be a genuinely better fit for GPU hardware than ray-marching an implicit field. It also produces a representation that's more directly inspectable and editable than a NeRF's opaque network weights, which matters for any pipeline that wants to validate or post-process the scene after training.

The limitations are ones a lot of subsequent research — including several papers reviewed below — set out to fix. Millions of independent, unordered Gaussians don't automatically correspond to a coherent surface, so converting a trained scene into a conventional mesh is nontrivial. Storage and memory can also become significant for large or detailed scenes, since Gaussian count grows with scene complexity. And, like NeRF before it, quality still depends heavily on having decent SfM-derived camera poses and sparse geometry to start from : a weak or noisy camera initialization can visibly degrade the final optimized scene.

[4] Review of "CityGaussian: Real-Time High-Quality Large-Scale Scene Rendering with Gaussians" **Authors: Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, Zhaoxiang Zhang (ECCV, 2024)**

The original 3DGS [3] formulation works beautifully on a single object or a room-scale scene, but scaling it up to a city block, a campus, or an entire neighborhood exposes a different set of problems entirely. A scene that large simply can't be optimized as one Gaussian set on one GPU — memory runs out, training slows to a crawl, and rendering an entire city's worth of Gaussians for every frame is wasteful when the viewer can only see a small part of it at once. This paper tackles exactly that scaling problem, in the same spirit that Mip-NeRF 360 [10] once tackled unbounded scenes for NeRF [9].

CityGaussian's [4] approach is built around a divide-and-conquer strategy paired with level-of-detail (LoD) rendering:

- The large scene is spatially partitioned into blocks, each optimized somewhat independently, which lets training be parallelized across multiple GPUs instead of forcing one massive joint optimization.
- A level-of-detail structure is built on top of the trained Gaussians, so distant or less-important regions are rendered with a coarser, cheaper representation while nearby regions keep full detail — directly analogous to LoD techniques in classical real-time graphics.
- The blocks and LoD levels are fused back into a coherent whole for rendering, so a viewer moving through the scene doesn't see seams between the independently optimized regions.

This kind of large-scale Gaussian pipeline is aimed at:

- City-scale and campus-scale digital twins, where the reconstructed area is far larger than a single-room or single-object 3DGS scene can practically handle.
- Real-time fly-through visualization of large environments — the explicit goal is to keep the interactive, real-time character of 3DGS even as scene size grows by orders of magnitude.
- Virtual production and simulation contexts (e.g., autonomous-driving simulation, urban planning) where large, explorable 3D environments are needed rather than a single bounded object.

The value of this work for a general-purpose media-to-3D pipeline is that it shows the core 3DGS representation doesn't have to be abandoned at scale — it can be adapted with fairly standard graphics techniques (partitioning, LoD) rather than needing a fundamentally different representation. That's an encouraging sign for future media-to-3D work that wants to stay agnostic about scene size rather than assuming everything fits in one optimization pass.

The trade-offs are mostly about added system complexity. Partitioning a scene well requires reasoning about where to draw block boundaries so that geometry and appearance stay consistent across them, and getting LoD transitions to be visually seamless is its own tuning problem — done poorly, both can introduce visible seams or popping artifacts as the LoD level changes. Training also now involves an extra orchestration layer (deciding how to split the scene, managing multiple parallel optimizations) that a single-scene 3DGS pipeline never has to deal with. Still, the direction is clear: as Gaussian Splatting matures, "does it scale past one room" has become as important a research question as "does it render quickly," and this paper is one of several recent answers to that question.

[5] Review of "A Survey on 3D Gaussian Splatting" Authors: Guikun Chen, Wenguan Wang (continuously updated arXiv survey, latest revision April 2026)

Rather than presenting one method, this is a living survey [5] that's tracked the 3D Gaussian Splatting field since almost the moment it appeared, and it's genuinely useful for this kind of review precisely because it keeps getting revised as the field moves — the latest version pulls in work through early 2026, including several directions that didn't exist yet when the original 3DGS paper [3] was published. The authors organize the fast-growing body of 3DGS research into clear taxonomies rather than just listing papers chronologically.

The survey's coverage spans several threads that matter directly to an integrated media-to-3D pipeline:

- Core theoretical foundations of the Gaussian representation and the rasterization-based rendering pipeline, framed as the baseline every later variant modifies.
- Efforts to move beyond the plain ellipsoidal Gaussian primitive itself — surface-aligned primitives that constrain splats to local tangent planes for better mesh extraction, alternative kernel shapes (bounded-support or heavy-tailed distributions) that handle sharp edges better than a smooth Gaussian can, and even non-Gaussian primitive redesigns that rethink the rendering operator entirely.
- Compression and efficiency techniques, since a naively trained scene can contain millions of Gaussians that are expensive to store and stream.
- Applications across robotics, urban mapping, autonomous navigation, and virtual/augmented reality, illustrating how far the representation has traveled from its original novel-view-synthesis use case.
- A survey like this is used for:
- Orienting a new project in a fast-moving space — rather than reading dozens of scattered papers, it gives a structured map of what's been tried and where the open problems are.
- Identifying which extensions of the base 3DGS representation (surface alignment, compression, large-scale handling) are relevant to a specific downstream goal, which is exactly how this review has used it to select the other papers discussed here.

What makes this particular survey valuable rather than just a bibliography is the honesty about its own limits — the authors describe the survey as "inevitably a biased view" given how young and fast-moving the field still is, and they focus deliberately on trends rather than claiming exhaustive coverage. That's a useful reminder for any literature review in this space, including this one: it is a snapshot of a field that is still actively rewriting its own assumptions every few months.

The main limitation, unavoidable for any survey of an active research area, is currency — new 3DGS variants appear on arXiv essentially every week, so even a continuously updated survey is always slightly behind the frontier by the time it's read. It's also, by nature, broad rather than deep — useful for identifying which specific papers (like the ones reviewed elsewhere in this paper) are worth reading in full, but not a substitute for reading them.

II. CONCLUSION

Looking back across these five papers, a fairly consistent story emerges, even though each one is solving a narrowly different piece of the puzzle. Camera and geometry recovery has quietly moved from a purely optimization-driven process — incremental SfM, careful geometric verification, bundle adjustment — toward feed-forward networks that regress poses and structure directly, with VGGT [1] and DUST3R [2] both showing that a well-trained transformer can now do in under a second what used to take an incremental pipeline minutes or hours. Neither paper claims classical SfM is obsolete; if anything, both position themselves as faster front ends that can still feed into, or be cross-checked against, more traditional geometric pipelines.

The representation side of the story is just as clear. 3D Gaussian Splatting [3] settled the argument between implicit neural fields and explicit, rasterized primitives largely in favor of the latter, and everything downstream — CityGaussian's [4] large-scale partitioning and the ongoing Chen & Wang survey's [5] taxonomy of alternative kernels and compression schemes both treat the Gaussian representation itself as a stable foundation to be extended, rather than something that needs replacing. Related feed-forward Gaussian-prediction methods point in a similar direction [6].

What ties all five papers together, for the purposes of an integrated media-to-3D pipeline, is a shared move away from treating reconstruction as a single monolithic step. Camera recovery, dense geometry, Gaussian optimization, and scaling to larger environments are increasingly modular, swappable stages — a pipeline can now choose a classical SfM front end [7] or a feed-forward one like VGGT [1], depending on the speed-versus-fidelity trade-off a given project needs. None of these methods, individually, spans the full distance from raw, unposed media to a validated, renderable, and optionally mesh-ready 3D scene. Read together, though, they lay out almost every piece such a pipeline would need — which is precisely the gap that future integrated pipeline work is positioned to close.

REFERENCES

1. J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "VGGT: Visual Geometry Grounded Transformer," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 5294–5306.
2. S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "DUST3R: Geometric 3D Vision Made Easy," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20697–20709.
3. B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.
4. Y. Liu, C. Luo, L. Fan, N. Wang, J. Peng, and Z. Zhang, "CityGaussian: Real-Time High-Quality Large-Scale Scene Rendering with Gaussians," in *Proc. European Conference on Computer Vision (ECCV)*, 2024.
5. G. Chen and W. Wang, "A Survey on 3D Gaussian Splatting," *arXiv:2401.03890*, 2024 (revised 2026).
6. L. Jiang, Y. Mao, L. Xu, T. Lu, K. Ren, Y. Jin, X. Xu, M. Yu, J. Pang, F. Zhao, et al., "AnySplat: Feed-Forward 3D Gaussian Splatting from Unconstrained Views," *ACM Transactions on Graphics*, vol. 44, no. 6, pp. 1–16, 2025.
7. J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
8. J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise View Selection for Unstructured Multi-View Stereo," in *Proc. European Conference on Computer Vision (ECCV)*, 2016.
9. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *Proc. European Conference on Computer Vision (ECCV)*, 2020.
10. J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details